TC1 Project: Know your customers

Aris Tritas Laurent Cetinsoy

December 13, 2016

Abstract

A collaborative filtering problem for recommandation was addressed Non Negative Matrix Factorization (NMF) techniques and with Autoencoders. The impact of rating temporal dynamic was studied by removing a proportion of the firsts and the last ratings of each customers. Similar model performances on the truncated dataset and the full one indicates that temporal information may not have a great impact.

1 Introduction

In collaborative filtering, items are rated by one or more members of a population. The aim is infer the missing ratings in order to recommend to the users, the items they would have given a high rating.

Let $R \in N^{n \times m}$ represent the rating matrix. Then r_{ij} is the rating given by the i^{th} user to the j^{th} item.

The dataset used contains 2536705 ratings of 3561 items given by 93705 users. Less than 1% of the rating matrix is filled.

2 Non negative matrix factorization

In non negative factorizations we seek SPD matrices U and V such that the rating product matrix can be factorized as $R = UV^T$. It is possible to find such matrices that minimize the reconstruction error defined below by using an iterative optimization method (e.g gradient descent):

$$\begin{aligned} U^*, V^* = \underset{U,V}{\arg\min} \frac{1}{2}||R - UV||^2 \\ + \alpha\lambda_1(||U||_1 + ||V||_1) \\ + \frac{1}{2}\alpha\lambda_2(||U||^2 + ||V||^2) \end{aligned} \tag{1}$$

The reconstruction error is computed over all non negative coefficients of the R matrix.

2.1 ALS-WR

The Alternating Least Squares with Regularization algorithm [?] minimizes 1.It does so by alternatively updating U while keeping V fixed and then updating V while keeping U fixed. The parameters used were $\lambda = 0.01$ and a latent dimension $d = 100$.

2.2 Temporal dynamics

In the context of matrix factorization, temporal dynamics can be achieved by adding bias terms in the objective function pertaining to either sides of the matrix factors. However, in the speicific problem there is no information with which to encode this bias.

3 Variational Auto-encoder

The auto-encoder is a neural network approach used in unsupervised learning for dimensionality reduction. It optimizes the following loss function:

$$\begin{aligned} L_{2,\alpha,\beta} = & \alpha(\sum_{j \in \mathcal{C}(\tilde{x})} [nn(\tilde{x}_j) - x_j]^2 \\ & + \beta(\sum_{j \notin \mathcal{C}(\tilde{x})} [nn(\tilde{x}_j) - x_j]^2) \end{aligned} \tag{2}$$

where $\mathcal{C}(\tilde{x})$ The first term is a loss over corrupted version of the inputs which the network seeks to denoise. The second term is the reconstruction loss.

In the supervised classification task, the auto-encoder's objective is to lower a prediction error over a training set.

The approach proposed by [?] tackles the collaborating filter problem. Two different auto-encoders can be trained depending on whether the network seeks learn a representation of users or items:

- U-CFN is defined as $\hat{\mathbf{u}}_i = nn(\mathbf{u}_i)$
- V-CFN is defined as $\hat{\mathbf{v}}_j = nn(\mathbf{v}_j)$

4 Learning rating sequences

In the context of the neural networks, the idea is to preserver sequence information so as to learn to predict the next watch. The authors of [?] predict the next item in the sequence instead of predicting a randomly held-out item. Specifically, for the auto-encoder, one approach would be to create multiple examples for each user by incrementally augmenting each input vector with entries from the rated items sequence and evaluating the network on the reconstruction of the future entry.

4.1 Testing the importance of the sequence information

There are different ways to test whether the sequences are important to the task of prediction: firstly, by destroying part of the sequences (in which case, the raw loss of information has to be balanced against the loss of succession). Secondly, by shuffling the sequence (in which case the loss in performance has to be balanced against the properties of the model w.r.t exchangeability).

5 Implicit side information

The test set provides an additional source of information, namely items which were seen but whose rating is unavailable for training. The model can use this information by conditioning the hidden units activations on both the visible units activations and a vector indicating which items were rated [?].

6 Results

The metrics used to evaluate the performance of each training method for this type of problem is either the RMSE (root mean squared error) or the MAE (mean absolute error). The scores below represent the RMSE and were computed on a validation set selected uniformly at random amongst the training set:

Model	ratings	k	lambda	RMSE
GD	500k	15	0.02	0.863
GD	All	15	0.02	0.951
ALS	500k	15	0.03	0.877
ALS	All	15	0.03	0.925
ALS	All	30	0.03	0.923
ALS	All	100	0.1	0.934
AE Item Vector	All	0	0	0.938
AE User Vector	All	0	0	0.913
AE User Vector	All	0	0.2	0.947

7 Conclusion

Different methods were compared in order to solve the problem of customer affinity prediction. When sparsity becomes an issue, no single method works best: exploiting the problem structure and combining multiple approaches has the potential of achieving the best results.