Collaborative Recommendation based on Variational Automatic Coding Machine

Zhiheng Zhang

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Zip code: 100000, China

Abstract. For the traditional collaborative filtering algorithm, only the scoring information cannot reflect the user's preference, which leads to low recommendation accuracy and easy over-fitting. This paper proposes an improved basic VAE model for film and television programs, adding auxiliary information as a priori of hidden variables. Distribution, will be the first method to use the heterogeneous auxiliary information in the VAE for a priori recommendation; at the same time, integrate multimedia information, add features such as pictures and texts, enrich the hidden variable space, and improve the recommendation effect; finally, open the data set. Experimental tests were performed on MovieLens. Compared with the PMF benchmark model, the algorithm is significantly better than the above method in the root mean square error index.

Keywords: VAE; Collaborative filtering; Matrix decomposition; Movie recommendation.

1. Introduction

With the continuous development of deep learning, the application of deep learning in many fields will improve the final effect. Deep learning algorithms such as variational self-encoders have shown great power in recommendation systems and many other fields, mainly because they capture user-project nonlinear relationships. As a variant of the self-encoder, the variational self-encoder performs very well in the field of unsupervised machine learning. In particular, the use of polynomial likelihood extension vae has proven to be very effective for collaborative filtering of implicit feedback, comparing polynomial possibilities with other commonly used likelihood functions in the collaborative filtering of latent factors, and showing superior to project recommendations. The latest baseline.

2. Variable Autoencoder

A variational autoencoder is a generation model that combines the idea of deep learning with statistical learning.Its initial idea is to automatically generate data from a distribution z. The variational autoencoder uses the variational Bayesian method, which performs efficient approximate reasoning and learning on the probability graph model, and involves the posterior Approximate optimization.

The basic idea of the variational self-encoder algorithm is: for each input data pointx_kThere is a hidden variable z_k To replace.Therefore, the final output x_k' Can be distributed by some probability$p_k(x_k|z_k)$Generated, this probability distribution is assumed to be a Gaussian distribution in this paper.The decoder function is capable of generating the final parameters of the generated distribution, and is itself constrained by a uniform set of parameters.At the same time, the encoder function is able to generate each probability distribution.$q_k(z_k|x_k)$The parameters are also constrained by uniform parameters.This encoding process is an abstraction of the input and is therefore also referred to as the recognition model.

Variational derivation points out that an approximate distribution needs to be found$p(z_k)$To replace the uncalculated distribution$q_k(z_k|x_k)$. The measure of similarity between two distributions is usually measured using kl divergence. so, the confidence lower bound objective function for a single input point is as follows:

$$L = -D[q_k(z_k|x_k)||p(z_k)] + E[\log p_k(x_k|z_k)]$$

The first term in the above equation is the kl divergence function, which can be regarded as a regularization term, and the second expectation can be regarded as the reconstruction error of the self-encoder. From distribution $q_k(z_k|x_k)$ Taking s samples in the sample and approximating the expectation of the samples will make the calculations simpler.

$$L = -D[q_k(z_k|x_k)||p(z_k)] + \int_{q_k(z_k|x_k)} \log p_k(x_k|z_k) dz_k \approx$$

$$-D[q_k(z_k|x_k)||p(z_k)] + \frac{1}{s}\sum_{1}^{s} \log p_k\left(x_k|z_k^{(s)}\right)$$

among them$z_k^{(s)}$Is one of the s samples.due top(z_k)Is a Gaussian distribution, so you can use re-parameter quantization to simplify the operation, ie for the sample$z_k^{(s)}$In this case, a small variable is obtained from the normal distribution but from the standard normal distribution, and then calculated using the expected sum and the standard deviation.$z_k^{(s)}$.At this point, all parameters can be optimized using a random gradient descent.

3. Matrix Decomposition

Matrix decomposition is a commonly used algorithm in collaborative filtering recommendation. The idea is to decompose the user-item scoring matrix.$R^{n\times m}$, respectively, get the user and item vector representation, and then get the user feature matrix$U^{k\times n}$Project feature matrix$V^{k\times m}$,among themnFor the number of users,mFor the number of projects,kThe hidden vector feature space dimension.The multiplication of the two matrices obtained by the decomposition is the predicted score:

$$R^* = U^T V$$

The algorithm optimizes the difference between the predicted score r and the true score.R^*Perform model training, that is, optimize the objective function:

$$E = I(R - U^T V)^2 = \sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(r_{ij} - u_i^T v_j)^2$$

among themI_{ij}To indicate the function,$I_{ij} = 1$Indicates that user i has scored item j, otherwise$I_{ij} = 0$; r_{ij}Indicates the rating of user j on item j;u_i as well asv_jThe feature vectors of user i and item j are respectively subjected to inner product operations, that is, predicted scores.Then use the convex optimization method to make its partial derivative zero, iteratively train the user and the project vector.

4. Model Establishment

In this model, We define the variable $u \in U = \{1 \dots m\}$ to index the users and $v \in V = \{1 \dots n\}$ To index the items. Partially observed user-item scoring matrix is defined as $R \in R^{m\times n}$. Every user $u \in U = \{1 \dots m\}$ Use the scores that can be observed for each item $R(u) \in (R_{u1} \dots R_A)$ As its description.The same; for each project $v \in V = \{1 \dots n\}$, we use the score of each user that can be observed $R(i) \in (R_{1i} \dots R_I)$ To show; we use$i_v \in (I_1 \dots I_n)$Indicates auxiliary information for the project,$i_u \in (I_1 \dots I_m)$Indicates the user's auxiliary information.Our goal is to design an item based VAE (user-based VAE), which represents the R(i)(R(u)) variable-divided self-encoders with lower-dimensional hidden layer vectors, and is also integrated into the auxiliary information of the project

(user). This is used as the prior distribution of the hidden layer space vector, and finally R(i)(R(u)) is reconstructed by the decoder. Figure 1 below represents item based-VAE for collaborative filtering.

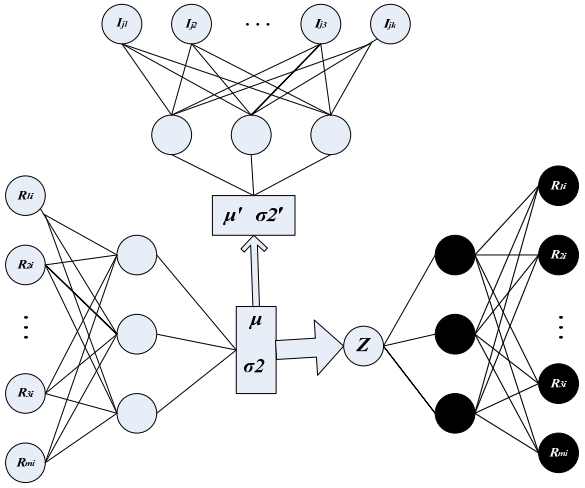

Figure 1. itembased-VAE model.

For each project, each user's rating vector $R(i) \in (R_{1i} \ldots R_I)$Input as endoder; project auxiliary information$i_v \in (I_1 \ldots I_n)$The prior distribution of the hidden vector as a variational autoencoder; the hidden layer vector Z is sampled according to the prior distribution of the auxiliary information; a new vector $R(i) \in (R_{1i} \ldots R_I)$It is coded by decoder.

5. Experimental Results

In this paper, the MovieLens 1M public data set is selected. In order to verify the overall performance of the proposed model and the comparison model, the data set is divided into training set and test set, 80% is the training set, and the remaining 20% is used for testing, and will be trained. The 20% of the set is randomly divided into verification sets for model parameter adjustment and does not participate in model training.The root mean square error (RMSE) is used as the evaluation standard for measuring the performance of the model.In the model training, the RMSE of the true score and the predicted score is minimized. The RMSE calculates the deviation between the predicted value and the true value, and squares the difference, and finally calculates the square root of the ratio of the magnitude N of the predicted data. The mathematical formula is as follows:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i,j}^{m,n}(r_{ij} - r'_{ij})^2}{N}}$$

among themnIndicates the number of users,mRepresents the number of movies, n is the number of ratings in the test set.r_{ij}Represents the true score of user i on movie j,r'_{ij}Represents the predicted score of user i for movie j.

The benchmark model PMF is compared with the model in terms of RMSE loss. Taking the MovieLens1M dataset as an example, in terms of iteration speed, the PMF can reach convergence in 100 steps of iteration, and the model can enter the convergence state after 50 iterations; In terms of degree, PMF eventually reached 0.98, and the model of this paper can reach about 0.85, which is significantly improved compared with the benchmark model. It can be seen that the model can effectively improve the performance of the model by adding auxiliary information of users and movies.

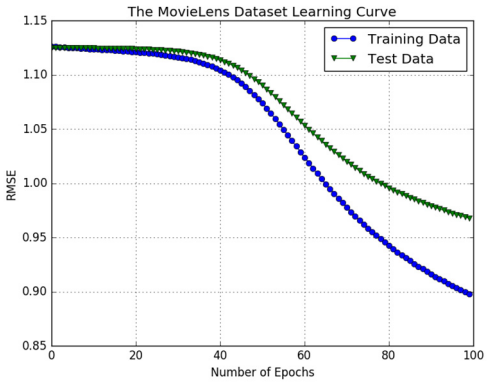

Figure 2. pmf-rmse loss graph

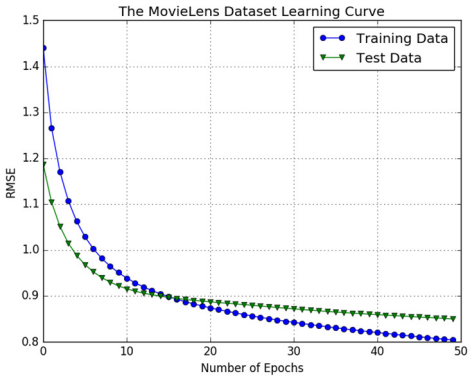

Figure 3. vae-rmse loss graph

6. Conclusion

This study proposes a new variational self-encoder model with the addition of multimedia information, which helps to improve the accuracy of missing data prediction through auxiliary information. Modeling from the user and project perspectives, respectively, replaces the original Gaussian priors that are unknown to the user or the project. These priors are empirically estimated as functions of the user or project auxiliary information.

References

[1]. RICCI F, ROKACH L, SHAPIRA B, et al. Recommender Systems Handbook [M]. Berlin: springer,2015.

[2]. Ajit P. Singh, Geoffrey J. Gordon. Relational Learning via Collective Matrix Factorization[J]. Acm Sigkdd International Conference on Knowledge Discovery & Data Mining, 2008, 40(46):650-658.

[3]. Paul Covington, Jay Adams, Emre Sargin. Deep Neural Networks for YouTube Recommendations[A]. Proceedings of the 10th ACM Conference on Recommender Systems [C], New York: ACM Press,2016.

[4]. Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al.2016. Wide & deep learning for recommender systems. In Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, 7-10.

[5]. Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 191-198.