Auto-Encoder based Recommendation Algorithm Combining Item Types

Tianci Wang

College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China.

Abstract

Traditional collaborative filtering algorithm relies on score matrix to generate prediction, fails to mine potential features related to item genres, which leads to a low recommendation accuracy. A denoising autoencoder based collaborative filtering recommendation algorithm combining item types is proposed. Firstly, combining item score data and genres data, the potential features of the items are extracted by denoising autoencoder. Then the prediction is generated by the collaborative filtering algorithm. Experiments on MovieLens dataset show that the new algorithm can mine the potential features of item genres comprehensively, and improve the recommendation accuracy.

Keywords

Denoising Autoencoder; Collaborative Filtering; Items with Same Genres; User Preference.

1. Introduction

With the advent of the information age, the explosive growth of data scale has brought about transformative development to human society, but has also brought about a serious problem of "information overload". It is difficult for people to select useful information from a large amount of data. The recommendation system came into being. The core of the recommendation system is the recommendation algorithm. Currently, the most widely used recommendation algorithm is collaborative filtering (CF) [1]. However, the traditional collaborative filtering algorithm only uses a sparse score matrix for recommendation, which can no longer meet the needs of users for effective information. For this reason, the personalized recommendation algorithm based on multi-information fusion has become a hot research topic [2].

When people browse the web aimlessly, they often narrow the scope of browsing by item types. The type of items can help people quickly locate the items they want. The item type not only reflects the characteristics of the item itself, but also reflects the similarities and differences with other items. The recommendation algorithm combined with the item type fully considers the relationship between the same type of items, making the recommendation result more real and complete. Frémal et al. [3] proposed a clustering algorithm based on item type weighting. Each cluster has its own rating prediction, and then uses a weighting strategy to merge these evaluations into the overall evaluation. SUO et al. [4] used item types to define the type similarity of two users, and then combined user-defined tags to define a new user similarity.

Cheng et al. [5] performed LDA clustering on item types to generate subject item clusters, and then generated prediction scores by LDA clustering through probability transfer. However, the above method only uses the explicit classification information of the item type, and does not combine the score information of the same type of items.

In response to the above problems, combined with the score vectors of the same type of items, a denoising autoencoder recommendation algorithm that integrates item types (GDAECF) is proposed.
2. Proposed Methodology

Taking into account the impact of item types on items, the GDAECF algorithm uses a denoising autoencoder to mine item implicit score features that contain item type information, and then calculate the item similarity, and finally the prediction score is generated through the collaborative filtering method. It is mainly divided into the following 3 steps:

1) According to the item-type matrix, obtain the set of items of the same type for each item, and then calculate the weighted average value of all item score vectors according to the user-item score matrix, as the item score vector of the same type of the item.

2) The original score vector of the item and the score vector of the same type of item are added to the dual-input dual-output denoising autoencoder to obtain the hidden features, and the item similarity is calculated according to the hidden features of the item.

3) According to the item similarity, the collaborative filtering method is used to generate the prediction.

The GDAECF algorithm flow is shown in Figure 1:

![Flowchart of GDAECF recommendation algorithm](image)

2.1. Score vector of the same type of item

items of the same type often have certain similarities with target item. Whether a item is favored by users, partly depends on some characteristics of the item itself, that is, personal characteristics; the other part depends on the type of the project, that is, common characteristics. Therefore, for the analysis of item characteristics, on the one hand, it is necessary to analyze the score vector of the item itself, on the other hand, it is also necessary to consider the score vector of the same type of items.
Given a set of users \( U = \{u_1, \ldots, u_n\} \), a set of items \( I = \{t_1, \ldots, t_m\} \), a set of types \( G = \{g_1, \ldots, g_l\} \), \( n, m, l \) represents number of users, items and types respectively.

In the score matrix, the rows represent the user’s score information, and the columns represent the item’s score information. The score matrix \( R \) is an \( n \times m \) matrix, and the element in the \( i \)-th row and \( j \)-th column is defined as formula (1):

\[
R_{ij} = \begin{cases} r, & \text{User } i \text{ has rated user } j \\ 0, & \text{User } i \text{ has not rated user } j \end{cases}
\]

Among them, \( r \) is the rating of user \( i \) on item \( j \).

In the type matrix, the rows represent the type information of the item, and the columns represent the item information of the type. The type matrix \( G \) is an \( m \times l \) matrix. The elements in the \( i \)-th row and \( j \)-th column are defined as shown in formula (2):

\[
G_{ij} = \begin{cases} 1, & \text{item } i \text{ belongs to type } j \\ 0, & \text{item } i \text{ does not belong to type } j \end{cases}
\]

The personality feature of item \( j \) is an \( n \)-dimensional score vector in score matrix; the same type score vector of item \( j \) is defined as formula (3):

\[
S^*_j = \frac{\sum_{i=1}^{m} (G_{ij} \cdot G^*_{i}) R_{ij} - (G_{j} \cdot G^*_{i}) R_{jj}}{\sum_{i=1}^{m} (G_{ij} \cdot G^*_{i}) - (G_{j} \cdot G^*_{j})}
\]

Among them, \( S^*_j \) is the score vector of same type of item \( j \), \( \cdot \) is the vector inner product, \( G_{is} \) is the type vector corresponding to item \( i \), and \( G_{js} \) is the score vector of item \( j \).

### 2.2. Similarity calculation based on denoising autoencoder

In order to obtain the item characteristics of the combining item type and express the influence of the same type on the item, the model adopts a dual-input and dual-output autoencoder, and uses a weighted hidden layer for fusion, as shown in Figure 2:

![Graph model of dual input dual output from DAE](image)

Figure 2. Graph model of dual input dual output from DAE

First, the denoising autoencoder adds noise to the score vector of the target item \( X \) and the score vector of the same type of items \( S \), so that a certain proportion of neurons do not work, \( \tilde{X} \) and \( \tilde{S} \) are obtained. And then encoders encode the two score vector to latent space, and the outputs of encode layer are computed by equations (4) and (5):

\[
h^X_m = f(W^X \tilde{X}_m + b)
\]

\[
h^S_m = f(W^S \tilde{S}_m + c)
\]

Where \( \tilde{X}_m \) and \( \tilde{S}_m \) are respectively the corrupted score data of item and the corrupter score vector of same type of items, \( h^X_m \) and \( h^S_m \) are the latent score feature of the target item and the
latent score feature of the same type of items. The parameters \( W^X, W^S, b, c \) are the weight parameters and biases of the network coding layer. \( f(\cdot) \) is sigmoid function. The superscripts X and S are used to distinguish the score vector of the target item from the score vector of the same type of items. \( F \) is the output of the weighted hidden layer, used to fuse the indirect effects of the same type of item, expressed as equation (6):

\[
F_m = ah^S_m + (1 - \alpha) h^X_m
\]  

(6)

Where \( \alpha \) is the weighting factor that balances the influence of the same type of items, and finally the original input data is reconstructed through two decoder layers. The two layers are represented by equations (7) and (8):

\[
\hat{X}_m = f(W^X F_m + b')
\]  

(7)

\[
\hat{S}_m = f(W^S F_m + c')
\]  

(8)

Where \( \hat{X}_m \) and \( \hat{S}_m \) are the score vector of the item \( X \) and the reconstruction vector of the score vector of the same type of item \( S \) respectively, and the parameters \( W^X, W^S, b', c' \) are the weight parameters and biases of the decoder layer, \( m \) refers to a certain item. The loss function is defined as formula (9):

\[
L = l(\hat{X}, \hat{X}) + l(S, S) + \lambda \Omega(W^X, W^S, W^X, W^S, b, b', c, c')
\]  

(9)

Where \( l(\cdot, \cdot) \) is the loss function for calculating the reconstruction error, \( \Omega(\cdot, \cdot) \) is the regularization parameter, and is the regularization term, defined as formula (10):

\[
\Omega(\cdot, \cdot) = \|W^X\|^2_P + \|W^S\|^2_P + \|W^X\|^2_F + \|W^S\|^2_F + \|b\|^2_F + \|c\|^2_F + \|b\|^2_P + \|c\|^2_P
\]  

(10)

By minimizing the loss function of the network, an encoder that can extract the hidden score features of the item is obtained. After the item score vector and the similar item score vector are provided to the encoder, the low-dimensional features of the item can be obtained, and then the Person similarity can be obtained. Pearson similarity formula is shown in equation (11):

\[
corr_{i,j} = \frac{\sum(F_{u,i} - \bar{F}_i)(F_{u,j} - \bar{F}_j)}{\sqrt{\sum(F_{u,i} - \bar{F}_i)^2 \sum(F_{u,j} - \bar{F}_j)^2}}
\]  

(11)

Where \( corr_{i,j} \) represents the Pearson similarity weight of the item \( i \) and the item \( j \), \( F_{u,i} \) represents the \( u \)th score feature of the item \( i \), \( \bar{F}_i \) represents the average value of the item \( i \) on all score features, and \( \bar{F}_j \) represents the average value of the item \( j \) on all score features.

2.3. Generate Prediction

After obtaining the similarity between the items and the time weight of the user to the item, the score prediction is made according to the traditional collaborative filtering idea, and the calculation method is shown in equation (12):

\[
P_{u,i} = \frac{\sum_{j=1}^{n_i} (R_{u,j} \times corr_{i,j} \times f(u, i))}{\sum_{j=1}^{n_i} corr_{i,j} \times f(u, i)}
\]  

(12)

Where is the predicted score of user \( u \) for item \( i \), \( n_i \) is the number of neighbors of item \( i \), is the time weight function, and is the Pearson similarity weight of item \( i \) and item \( j \).

3. Experiment and result analysis

In order to verify the effectiveness of the algorithm GDAECF in this paper, comparative experiments were carried out to analyze the function of each module of the algorithm and verify the performance of the algorithm.
3.1. Experiment data and experiment environment

The experiment uses the MovieLens 100k data set, which contains the type of movie. A total of 943 users rated more than 100,000 of 1682 movies. 80% of them were randomly selected as the training set and the remaining 20% as the test set. In order to reduce the experimental error, the experiment was carried out by means of five-fold cross-validation.

The experiment environment is: Inter Core i5 processor, 16G memory, Windows 10 x64 operating system, and python 3.6 is used to implement the code.

3.2. Performance Metric

Mean Absolute Error (MAE) is one of the evaluation criteria commonly used in recommender systems. The accuracy of prediction is judged by calculating the average error between the predicted score and the actual score, as shown in formula (13):

\[
MAE = \frac{\sum_{i=1}^{N}|r_{ui} - \hat{r}_{ui}|}{|N|}
\]

Where \(r_{ui}\) is the predicted score of user \(u\) on item \(i\), \(r_{ui}\) is the actual score of user \(u\) on item \(i\), and \(N\) is the number of predicted items. The smaller the MAE, the better the accuracy of the prediction.

3.3. Parameter Selection

In the recommendation algorithm, the selection of related parameters has an impact on the accuracy of the recommendation. Refer to previous research results on MovieLens dataset [6], the number of hidden layer units \(H\) is set as 400, and the value of \(\lambda\) is set to 0.0001.

The weighting factor \(\alpha\) is to balance the influence of the same type of items, and the value is in the range of \([0,1]\), as shown in Figure 4. In the experiment, the parameter \(\alpha\) is set to 0.1, 0.2, ..., 0.9, where \(\alpha=0\) and \(\alpha=1\) means that the score prediction is based only on the same type of project and only based on the project itself. The larger \(\alpha\) is, the greater the influence of the same type of items, the smaller the \(\alpha\) is, the smaller the influence of the same type of items.

Figure 3 shows that if the weight factor \(\alpha\) is too large or too small, the recommendation accuracy of the algorithm is not very good, and when it is set to 0.4, the accuracy of the recommendation algorithm is the best.
3.4. Comparison Experiment

In order to verify the effectiveness of the denoising autoencoder recommendation algorithm (GDAECF), the following three algorithms are selected for comparison: the traditional item-based collaborative filtering recommendation algorithm (ICF) [7], item clustering based on weighting algorithm (WS) [3] and auto-encoder recommendation algorithm ignoring item type factors (DAE). Under different number of neighbors $N$, the MAE changes of different algorithms are shown in Figure 4.

![Figure 4. Changes of MAE value under different N value](image-url)

Figure 4 shows that the performance of the GDAECF algorithm is significantly better than the ICF algorithm, the WS algorithm and the DAE algorithm. When the number of neighbors is too low, the performance of the GDAECF algorithm and the WS algorithm are roughly equivalent, but when the number of neighbors is greater than 20, The performance of the GDAECF algorithm is significantly better than the WS algorithm. This is because when the number of neighbors is small, the WS algorithm uses weighting strategy to provide a more accurate prediction. As the number of neighbors increases, GDAECF also uses a denoising autoencoder to extract items features that contain the same type of item score data. The latent feature makes the judgment of neighboring items more reasonable overall. Therefore, the accuracy is also improved with the increase of the number of neighbors. This shows that mining the hidden features of the same type of item ratings can improve the performance of the recommendation.

4. Conclusion

Based on the research of item types, this paper incorporates the impact of the same type of item scores in the calculation of item similarity, and proposes a denoising autoencoder recommendation algorithm that integrates item types, fully excavated the potential characteristics of the same type of items, and improved the accuracy of prediction. In the future, we can study factors of user interest changes, try to explore hidden features related to time, explore more reasonable weighting factors, and make the recommendation algorithm more humane.
References


