Design and Implementation of Movie Recommendation System Based on Naive Bayes

Liu Shuxian 1*, Fang Sen 1

1 School of Information Science and Engineering, Xinjiang University, Urumqi, Xinjiang Province, 830046, China
2 Corresponding author’s e-mail:lsx623@163.com

Abstract. Today, with the rapid development of the Internet industry, the era of big data has arrived. What follows is that the surge in data volume has brought great challenges to people's rapid and timely screening of useful information. Based on this highly realistic problem, the recommendation system that solves the user's individual needs is born. In this paper, based on the theory of Naive Bayes, the MovieLens data set is used for testing. The user's scoring data are used for similar analysis to generate the user's similarity matrix, and the target user can be personalized. Compared with other algorithms, this paper has a good recommendation effect.

1. Introduction

Throughout the major Internet industries, whether it is the film and television industry, music platform or e-commerce platform, the amount of data information of each platform has increased dramatically. Only by extracting the effective information and providing the most useful information to users can we better increase business interests. Excavating the information that the user needs can not only improve the user experience, but also bring convenience to the user's life, and can increase the user's purchasing tendency, thereby achieving the effect of sales promotion. At present, the data on the network are too complicated and redundant. In addition, most users have shifted from physical store consumption to online shopping [1]. On the issue of this information overload, although the traditional search engine can achieve information indexing, it cannot be recommended according to the user's preference. Especially with the advent of the era of big data, information overload exists in all major industries of the Internet, and it has been unable to meet the individual needs of users. For example, search engines cannot make intelligent recommendations based on user preferences, for the input of the same keywords, returned are unified search results, can not meet the user's individual needs [2]. In this case, the recommendation system [3] that solves the user's individual needs is born. According to existing user information, such as interest preferences, collection information, browsing footprints, etc., the information of interest to the user is recommended, thereby achieving the effect of sales promotion. For example, Recommended rankings for major theaters such as cat's eye, personalized recommendations after user login, etc. A personalized recommendation system based on user preferences can greatly enhance the user's experience and purchase requirements, thereby increasing commercial sales profits.

There have been some problems with the personalized recommendation system developed so far, such as cold start and data sparse [4]. For cold start, generally use non-personalized recommendation, and then wait until the collected user data reach a certain amount, then carry out personalized recommendation. With the advent of the era of big data, the amount of data has become larger and
larger, and the proportion of data evaluated by users is too low, resulting in extremely poor
recommendation. For this problem, similar users can also be clustered, and users who are grouped into
the same class can be regarded as neighbors of the user, and then the behavior of the target user can be
predicted according to the preference of the neighbor. User-CF, Mb-CF, and Item-CF are typical
representatives of collaborative filtering algorithms. With the development of the recommendation
system, the real-time recommendation system has become a research hotspot. The real-time
recommendation system combines data mining, machine learning, natural language processing, and
behavioral psychology.

In the final analysis, the recommendation system is a problem of classification and regression. In
terms of classification, combined with the machine learning method, the naïve Bayesian classifier has
a simple classification model and convenient operation, and is considered to be the top ten data mining
algorithms [5]. The naïve Bayesian classification is irreplaceable in some classification fields, and
multiple classifications can be obtained [6]. The naïve Bayesian algorithm is established under the
condition that the attribute conditions are independent of each other, and the training data set is used to
estimate the probability items of the classification. This paper uses Naive Bayes as a recommendation
algorithm to recommend movies of interest to target users, with good recommendation results.

2. Naive Bayes-based recommendation algorithm

The naïve Bayesian classification algorithm has high classification accuracy when dealing with a large
amount of data, and the algorithm is mainly used to predict the unrated data. The idea of Naive Bayes
is the problem of probability classification. Under the condition that the user-specified feature items
are satisfied, the items to be classified are given, and the probability of occurrence of each category
under the condition of this occurrence is obtained[7], and then to be the classification items are
classified into the categories with high probability of seeking, and are also used as the result of
prediction. For example, according to the historical data of a certain user watching a movie in a movie
theater or on the Internet, it is found that five films satisfy the condition under the premise of
satisfying various conditions of the user, and the first movie has the highest probability of occurrence,
according to the naïve Bayesian algorithm, recommend the first movie with the highest probability of
occurrence to this target user.

Based on Naïve Bayesian algorithm, the implementation process can be roughly divided into three
stages.

1. Appropriately dividing each feature attribute of the movie, and then forming the item to be
classified into a training set. The classification process is an important part of the whole system, which
directly affects the effect and quality of the recommendation. The quality of the classification mainly
depends on the construction and division of the characteristic attributes.

2. Calculate the frequency of occurrence of each category in the training sample and the
conditional probability estimate of each category for each category, and record the calculated result.
The input of the process is the feature attribute and the training attribute, and the output is the
classification. [8].

3. Using the classifier to classify the classified items, and obtain the category to which the items to
be classified belong.

In the naïve Bayesian algorithm, the user has a label for each movie, likes and dislikes, and then we
use the data to train a classifier to divide whether each movie in the data set is a category that the user
likes, with this algorithm, we can recommend movies that users like. In short, the naïve Bayes-based
classification needs to train the classifier first, and then calculate the probability that the user likes the
movie based on the training data of the training classifier, and recommend the movie with high
probability, and the calculation is as shown in Equation 1.

\[
P(b_i | a) = \frac{P(a | b_i)P(b_i)}{P(a)}
\]  

(1)
3. Naive Bayes classifier

In the big number theorem, if the sample data are independent of each other and are subject to the same distribution of random variable sequences, the frequency of the event can be used instead of the probability of the event. Similarly, under the condition of naive Bayesian theory, that is, the values between each attribute of the sample are independent of each other, the frequency of occurrence of each sample can be used to calculate the prior probability of the sample data in the data set.

The problem is concentrated on the calculation of the posterior probability \( P(A|B) \). Since the sample data A is not a single attribute in the data set, this paper uses the MovieLens data set, A represents the user's data information, and the user's data information includes not only the name attribute, but also data such as gender, age, occupation, and favorite movie type, that is, A represents a plurality of data attributes and constitutes a user's attribute vector. In the calculation of the posterior probability, the sample to be taken contains all possible combinations of attributes. If there are no attribute conditions independent of each other, all probability values are zero. In the Naive Bayes classification, it is pointed out that the attribute conditions of the sample are independent. Under this assumption, the posterior probability is calculated as shown in Equation 2. In this formula, the individual values of the attributes are independent of each other.

\[
P(a|b) = \frac{P(a)P(b|c)}{P(b)} = \frac{P(a)}{P(b)} \prod_{i=1}^{d} P(b_i|a)
\]

3.1. Naive Bayes classification criteria

Under the condition that the values of the characteristic attributes of the samples are independent of each other, assuming that \( x \) is a feature quantity, after calculating the conditional probability of the category, the item to be classified belongs to the category with the highest probability. The classification criterion of Naive Bayes depends on its classification idea. In the recommendation system, it is necessary to analyze the data such as the score of the user by the user, and after training the classifier, recommend the movie with a higher probability to the target user. The classification criteria can be as shown in Equation 3, also as a prediction function.

\[
y = f(x) = \arg \max_{c \in k} P(c) \prod_{i=1}^{d} P(x_i|c)
\]

In the equation, \( \prod_{i=1}^{d} P(x_i|c) \) represents the product of multiple attributes of the sample. These attributes are independent of each other, and the values are independent of the attributes. The \( c \) in \( P(c) \) is the maximum value that needs to be calculated.

3.2. Naive Bayes classification process

![Figure 1. Classification process.](image-url)
The basic theory of naive Bayesian classification is very simple, but it is the simplest and most efficient classification algorithm in some fields. For the items to be classified that have been given in the data set, calculate the probability of occurrence in each category in this category, and then classify the item to be classified as the highest probability. The classification process is shown in Figure 1.

3.3. Naive Bayes' Laplacian Smoothing
Before the naive Bayes classification, the recommendation system needs to process the user's rating, and the user's rating is 1-5. This requires dividing the user's high score into the user's favorite, and the low score is classified as a category that the user does not like, but if the user has not scored one or some of the movies, it is easy to generate a zero probability problem when calculating the data. The zero-probability problem means that when calculating the probability of the sample to be tested, the training set does not have the value of the sample attribute, which results in the sample probability value being zero. From a mathematical point of view, the mean value of the central limit theorem is zero.

To solve the zero-probability problem, Laplace smoothing is defined as follows: When the training samples are large enough, the estimated probability change caused by adding one to each component count is negligible. In numerical value, it needs to be processed on the mean value obtained by the central limit theorem. The numerator and the denominator are simultaneously added with one number. The numerator generally adds 1 to the treatment, and the denominator adds the classification number. The code is implemented as follows.

```python
def laplace_smoothing(numerator, denominator, a=1):
    """
    Laplace smoothing
    :param numerator: Molecules that calculate probability
    :param denominator: Calculate the denominator of probability
    :param a: Laplace smoothing parameter
    :return: Probability after Laplacian smoothing
    """
    return (numerator + a) / (denominator + 2 * a)
```

4. Performance evaluation index
In any evaluation model, if performance comparison is to be performed, a unified evaluation index needs to be established. In the personalized recommendation system, a number of recommendation algorithms can generally be implemented, such as content-based recommendation, collaborative filtering-based recommendation, or naive Bayes-based recommendation system using machine learning. For the recommendation system implemented by different algorithms, the performance of the system will not be the same, the recommendation results are also different, and the system operation efficiency also has more or less differences. Then the performance of the system needs to be evaluated in order to select the recommended algorithm suitable for the system, so that the efficiency and accuracy of the recommendation can be greatly improved. In this paper, the performance of the system is evaluated mainly by four evaluation indexes: accuracy, precision, recall rate, and F1-score.

Before the evaluation index calculation, according to the general method of data mining theory, the most widely used evaluation model prediction ability is the two-dimensional confusion matrix, and the two-dimensional confusion matrix table is shown in Table 1.

| Use category | forecast result       |
|--------------|-----------------------|
| Category 1 (used) | True Positive, TP     |
| Category 2 (not used) | False Positive, FP    |
| Category 1 (recommended) | False Negative, FN   |
| Category 2 (not recommended) | True Negative, TN      |
Among them, TP refers to the actual positive sample, the predicted positive sample is recommended. FP refers to the actual negative sample, the predicted positive sample, is recommended; TN refers to the actual negative sample, the predicted negative sample, is not recommended; FN refers to the actual positive sample, and the predicted negative sample is not recommended[9].

4.1. accuracy
Accuracy refers to the proportion at which the positive sample is predicted to be correct, as calculated by Equation 4.

\[
\text{precision} = \frac{TP}{TP + FP} \times 100\%
\]  

(4)

4.2. precision
The precision rate indicates the proportion of the test set that is recommended for the user, and the calculation is as shown in Equation 5.

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \times 100\%
\]  

(5)

4.3. recall rate
The recall rate refers to how many movies are recommended in the list of movies that the user likes, and the ratio of the number of positive examples of the correct classification to the actual number of positive examples is calculated as shown in Equation 6.

\[
\text{Recall} = \frac{TP}{TP + FN} \times 100\%
\]  

(6)

4.4. F1-score
F1-score combines the recall rate and the accuracy rate for evaluation, and the calculation is as shown in Equation 7.

\[
F1\text{-score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \times 100\%
\]  

(7)

5. System architecture of the movie recommendation system
The film recommendation system of this paper is divided into three layers: data layer, data processing layer and display layer, as shown in Figure 2.
The data layer mainly stores data information of the movie such as movie id, movie name, movie type, etc., user data information such as user name, age, occupation, etc., rating information such as user id, movie id, rating. In order to improve the processing efficiency of the data, the data of the data layer are mainly stored in csv files.

The data processing layer is also the recommendation engine, which submits the user features and movie features extracted from the data layer to the recommendation engine. The recommendation engine then uses naive Bayes' recommendation algorithm for data processing to rank the recommended movie results, and then recommend The results are passed to the front-end interface for presentation processing.

The display layer uses Tkinter to process the recommended results, and the GUI application can visually see the information of the recommended movie.

In the development process of the system, PyCharm is used as a development tool and developed in Python language.

6. Implementation of movie recommendation algorithm

The system uses the MovieLens data set. When the user scores information processing, the user is scored 4 points or more as a high score, and the rest is regarded as a low score, and then divided into categories that the user likes and dislikes. Then use the naive Bayesian algorithm to calculate the probability that each movie belongs to the target user's favorite type, and finally recommend it. The algorithm implementation process is as follows.

```python
for new_rating_movie in new_ratings_info:
    same_rating_time = 0
    high_score_time = 0
    low_score_time = 0
    for rating in same_user_ratings_info:
        rating_movie_id = [item[1] for item in rating]
        rating_score = [item[2] for item in rating]
        if movie_id in rating_movie_id and new_rating_movie[1] in rating_movie_id:
            flag1 = 0
            flag2 = 0
            if rating_score[rating_movie_id.index(new_rating_movie[1])] >= '4':
                flag1 = 1
            if new_rating_movie[2] >= '4':
                flag2 = 1
            if flag1 + flag2 == 0 or flag1 + flag2 == 2:
                same_rating_time += 1
            if rating_score[rating_movie_id.index(movie_id)] >= '4':
                high_score_time += 1
        else:
            low_score_time += 1
    high_score_rating_PR *= laplace_smoothing(high_score_time, same_rating_time)
    low_score_rating_PR *= laplace_smoothing(low_score_time, same_rating_time)
```

7. Film recommendation performance evaluation

7.1. Analysis of performance indicators

In this system, the number of recommended movies is limited to 10, and 4-6 users are randomly selected to perform data testing, and their accuracy, precision, recall rate, F1-score are obtained respectively, and comparative analysis is performed. The specific data of three of them are selected as shown in Table 2 and Figure 3.
| Table 2. User’s evaluation index calculated value |
|---|---|---|---|
| accuracy | precision | recall | f1 |
| 0.7234 | 0.8918 | 0.78571 | 0.8354 |
| 0.5772 | 0.6075 | 0.6956 | 0.6486 |
| 0.5 | 0.7 | 0.6 | 0.6461 |

From Table 2 and Figure 3, it can be seen intuitively that the precision of the naive Bayes algorithm used in this paper is basically maintained at around 0.6, at the same time, the system has a relatively high accuracy rate and it is considered that users have a high probability of clicking on the recommended movies, equivalent accuracy and recall rate, as can be seen from the F1-score data, the overall evaluation is maintained at around 0.7, recommended movies that users may be interested in, and in these recommended movies, there are very mostly opted by users. Overall, the naive Bayes-based recommendation system has good performance in recommending.

7.2. Comparison of different algorithm performance indicators

At present, the more popular recommendation algorithms are Naive Bayes algorithm, Item-CF based on collaborative filtering, User-CF, and so on. The system compares the performance of Naive Bayes, Item-CF, and User-CF in the recommended performance, as shown in Figure 4.

![Figure 4. Recommended performance of different algorithms on the same data set](image-url)
Several popular algorithms, Naive Bayes, Item-CF, and User-CF, were tested in the same training set MovieLens dataset, and performance comparisons were made from accuracy and recall. It can be seen from Fig. 4 that the naive Bayesian algorithm has higher accuracy in recommending the film compared to the collaborative filtering based algorithm. From the recall rate based on naive Bayes, it can be seen that it is recommended for users. In the movie, the user has a high possibility to click to browse. Comprehensive evaluation, based on Naive Bayes' recommendation algorithm has better recommendation performance.

8. Conclusion
This paper applies naive Bayesian recommendation algorithm, uses the characteristics of movie data in MovieLens dataset and user's preference information on movies to describe the user's hobbies in movie selection, and then finds movies that meet the interests of target users and recommends them. According to some user recommendation results randomly selected, the naive Bayesian recommendation algorithm has a good recommendation effect.

Acknowledgement
Project funding: National Natural Science Foundation of China (61762085)

References
[1] Shi Yingxi. (2018)Research on personalized recommendation method for short text sentiment analysis. Harbin Engineering University, Harbin.
[2] Shan Jingjing. (2015)Research on content-based personalized recommendation system. Northeast Normal University, Changchun.
[3] Walter F E,Battiston S,Schweitzer F. (2008)A model of a trust-based recommendation system on a social network.Autonomous Agents and Multi-Agent Systems, 16(1): 57-74.
[4] Li Daxue, Xie Mingliang, Zhao Xuebin. (2010) Collaborative filtering recommendation algorithm based on naive Bayesian method[J]. Journal of Computer Applications, 30(6): 1523-1526.
[5] Wu X,Kumar V,Quinlan J R,et al. (2008) Top 10 algorithms in data mining. Knowledge & Information Systems, 14(1): 1-37.
[6] Liu Ying. (2007)Analysis of the Application of Bayesian Method in Text Classification. Computer Knowledge and Technology, 4(22): 1074-1076.
[7] Huang Yuda, Wang Yiran. (2012)Decision Tree Classification Based on Naive Bayes and ID3 Algorithm. Computer Engineering, 38(14): 41-43.
[8] Zhang Xin, Ma Yong, Cao Peng. (2012)Method of Trojan horse traffic identification based on Bayesian classification algorithm. Information Network Security, 8: 115-117.
[9] Zhang Jingjing. (2015)Research on Support Vector Machine Model and Algorithm Based on Unbalanced Data Set. China Agricultural University, Beijing.