http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (13): 2611-2616, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.2611.2616
© 2013 Asian Network for Scientific Information

Apply of Cloud Mining and Taste Algorithm in Network Video Recommendation

Xin Wang
Computer Engineering College, Weifang University, Shandong, 261061, China

Abstract: Recommendation engine algorithm is collaborative filtering and personalized recommendation technology which, by viewing the habits of a large user groups, picks out the most extensive users who will choose to continue watching after watching a video and recommends to the new users. This study designs the cloud mining architecture and technology implementation in the network video applications, gives the cloud mining formalized algorithm and analyzes the network video programs by using collaborative filtering algorithm based on content. This algorithm should be the best algorithm for cloud mining in the recommended application of network video.

Key words: Cloud mining, recommendation engine, collaborative filtering algorithm, taste algorithm

INTRODUCTION

More and more people need to find what they want from the vast amounts of information and the search engine is the best way to quickly find the target information. However, the search engine does not completely meet the needs of users to find the information, because, in many cases, the users are not clearly aware of their needs, or their needs are difficult to use simple keywords to be described. In other case, they need to conform to the result which they like personally. Therefore, the recommendation system is presented, corresponding with the search engine which is used to be called as a recommendation engine (Van Beek *et al*., 2003). With the continuous growth of data and content on the Internet, people begin to pay more attention on the role of the recommendation engine in Internet applications. The recommendation engine can analyze users' behavior to predict their preferences, so that they can more easily find their potential information from the Internet.

DEVELOPMENT OF NETWORK VIDEO RECOMMENDATION AND CLOUD MINING

Video information through the network is now becoming increasingly popular. In order to better meet the needs of users, as well as to provide better service, the video recommendation engine has been widely applied. This means that users can still easily find favorite content without knowing of video content which cannot be searched. This technology establishes the interest mode through the movie-viewing habits and records of a large user groups and personalizes video recommendation for user preferences. The recommendation engine algorithm is collaborative filtering and personalized recommendation technology that by the viewing habits of a large user base, picks out the most extensive users who will choose to watch after watching a video and recommends to the new user.

During watching wonderful TV programs in Network video, demand for advertising is becoming the focus of attention. In order to make video ads watching audience grow steadily, the main task of the video website is trying to provide the programs preferred by the users. The users in the process to watch a favorite video will also see the ads which results in economic benefits (Van Beek *et al*., 2003). In this context, data mining technology comes into being.

Why is cloud computing used on massive data mining? First of all, the data content is becoming massive today and a high-performance machine or more large-scale computing device is required. In fact, in order to get comprehensible knowledge from massive amounts of data, the large-scale data mining is a prefer approach. With the particularly fast growth of data on the Internet, data mining task becomes more complex than search task (Wei *et al*., 2011). There are some specific aims in mass data mining which led to the need to have a good development and application environment in the mining process. In this case, the approach based on cloud computing is preferred.

CLOUD MINING ARCHITECTURE AND ALGORITHMS

Cloud mining architecture is shown in Fig. 1, in the bottom of the structure the Hadoop framework is used as the data sources of cloud mining. The users' viewing records are used as the input of the Mahout DataModel component. The final outputs to the users are the

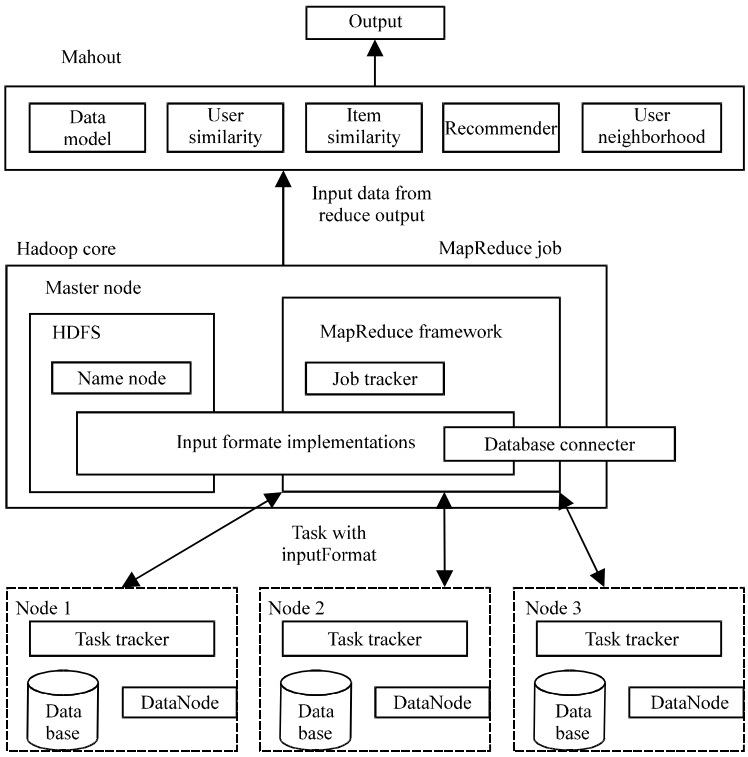

Fig. 1: Cloud mining structure with Hadoop as data sources

recommended video programs. Open source Mahout Project could be used into cloud mining machine algorithm. Apache Mahout is a new open source project developed by Apache Software Foundation (ASF) and its main goal is to create scalable machine learning algorithm for developers free to use under the Apache License (Owen *et al*., 2011). The project has developed into its second year with only a public release up to now. The Mahout contains many implementations, including collaborative filtering, clustering and classification, etc. In addition, by using the Apaehe Hadoop library, Mahout can be effectively extended to the cloud.

Firstly, the Hierarchical Bayesian clustering algorithm is utilized to analyze user behavior data and to achieve personalized recommendation function, through which the recommended program's ratings probability is estimated and then the estimated value is sorted from high to low. And then the sorting front programs are recommended to the users.

Bayes' theorem solved the problems often encountered in real life. That is, given a conditional probability, how to get the probability of exchanged two events. It is the case of knowing P(A|B) how to obtain P(B|A).

P (A|B) means the probability of occurrence of event A under the premise of the event B has occurred; it is called the conditional probability of event A given event B. The basic formula is:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad (1)$$

Bayes' theorem is useful because P(B|A) is usually difficult to be obtained, although P(A|B) could be easily obtained directly. However, P(B|A) is more concerned for us and Bayes' theorem provides us an effective way from P(A|B) to P(B|A). The Bayes' theorem is given by:

$$P(B|A) = \frac{P(A|B)P(B)}{P(A)} \quad (2)$$

HBC (Hierarchical Bayesian clustering) is a typical agglomerative hierarchical clustering algorithm and it makes the posterior probability as a maximum objective function to obtain a good clustering result (Tun and Thein, 2007).

Two categories are combined into one category in every step of the HBC algorithm. The choice is based on

the fact that the combined posterior probability P(C|D) is largest that is, every step of the local optimization is objective function with respect to P(C|D). Where D is a set of documents denoted by $D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$. Classification scheme C represents a set of classes which is a partition of D and denoted by $C = \{c_1, c_2, \ldots, c_i, \ldots, c_m\}$, $c_i \in D$, $c_i \cap c_j = \Phi$, $\forall i \neq j$.

In the initial stage of clustering, each document is treated as a separate category that is, $c_i = \{d_i\}$. At this point, the classification scheme is C_0. After the first k steps are completed, the classification scheme is C_k. The best clustering scheme C_{k+1} is to choose two appropriate categories c_x and c_y merged:

$$P(Ck|D) = \prod_{c\in C}\prod_{d\in c} P(c|d) = \prod_{c\in C}\prod_{d\in c} \frac{P(d|c)P(c)}{P(d)} = \frac{\prod_{c\in C} P(c)^{|c|}}{P(D)} \prod_{c\in C}\prod_{d\in c} P(d|c) = \frac{PC(C)}{P(D)} \prod_{c\in C} SC(c) \quad (3)$$

Where:

$$PC(C) = \prod_{c\in C} P(c)^{|c|}$$

$$SC(c) = \prod_{d\in c} P(d|c)$$

Between C_k and C_{k+1}, $C_{k+1} = C_k - \{C_x, C_y\} + \{C_x \cup C_y\}$ is satisfied and thus:

$$\frac{P(C_{k+1}|D)}{P(C_k|D)} = \frac{PC(C_{k+1})}{PC(C_k)} \frac{SC(c_x \cup c_y)}{SC(c_x)SC(c_y)} \quad (4)$$

For the step k+1, $P(C_k|D)$ is a known constant, therefore, it does not need to directly calculate $P(C_{k+1}|D)$ to find the best c_x and c_y. The first approximate value of the above formula is:

$$\frac{PC(C_{k+1})}{PC(C_k)} \propto A^{-1}$$

and A is a constant more than one. By using the Bayesian theory the conditional probability can be calculated as follows:

$$P(d|c) = P(d) \sum_t \frac{P(T = t|d)P(T = t|c)}{P(T = t)} \quad (5)$$

where, T = t means the event which characteristic word is exactly equal to t.

Now the expression of:

$$\frac{P(C_{k+1}|D)}{P(C_k|D)}$$

is achieved. And:

$$(\hat{c}_x, \hat{c}_y) = \arg\max_{(c_x, c_y)} \frac{P(C_{k+1}|D)}{P(C_k|D)}$$

which is the two categories to maximize $P(C_{k+1}|D)$. So the k+1 step should merge $\hat{c}_x$ and $\hat{c}_y$ into one category.

Summarizing the above analysis, the formal algorithm can be drawn as follows:

Input:
$D = \{d_1, d_2, \ldots, d_i, \ldots, d_n\}$: contain n documents of the input data
Initialize:
$C = \{c_1, c_2, \ldots, c_i, \ldots, c_n\}$, $c_i = \{d_i\}$, $1 \le i \le n$
For all $c_i, 1 \le i \le n$ calculate the $SC(c_i)$
For all $c_i, c_j, 1 \le i \le n$ calculate the $SC(c_i \cup c_j)$
For k = 1 to n-1 do
$(\hat{c}_x, \hat{c}_y) = \arg\max_{(c_x, c_y)} \frac{SC(c_x \cup c_y)}{SC(c_x)SC(c_y)}$
$C_k = C_{k+1} - \{\hat{c}_x, \hat{c}_y\} + \{\hat{c}_x \cup \hat{c}_y\}$,
For all $c_i, c_j, 1 \le i \le n$ in C_k calculate the $SC(c_i \cup c_j)$
Function SC(c)
Return $\prod P(d|c)$

The HBC algorithm has been tested and compared with Single-link Method and Ward's Method. In the results of clustering the data set containing 1252 articles, the accuracy of the Single-link is up to 62%, Ward's Method is close to 70% and HBC receives a higher accuracy rate of 74%. And hereby the HBC algorithm can make a very significant improvement in the accuracy of the clustering.

TASTE ALGORITHM IN NETWORK VIDEO RECOMMENDATIONS

The video recommendation system carries on the separate collection by the users' watching records on the bottom. And the collaborative filtering algorithm Taste which Apache Mahout provided to calculate and analyze the classified user view data, is used to identified the current user's viewing preferences, find the users with the same viewing hobbies and recommend to the current users the information about which users often watch the video program. It is a Java-based, scalable, efficient recommendation engine (Hongtao *et al*., 2011). Taste not only is the most basic user-based and content-based recommendation algorithm but also provides extension interface, with help of which users can easily define and implement the recommendation algorithm (Gelbard *et al*., 2007). At the same time, Taste can not only apply to Java applications but also it can be used as a component of the internal server to the outside with the recommended logic in the form of HTTP and Web Service (Sambasivam and

Theodosopoulos, 2006). The design of Taste enables it to satisfy the enterprise's requirement of recommendation engine in performance, flexibility, extendibility and so on. The Taste consists of the following five major components (Chaturvedi *et al*., 2001).

DataModel: DataModel is the abstract interface for the user's preference information and its specific implementation supports user preference information extracted from any type of data source. The Taste defaults to provide JDBCDataModel and FileDataModel, with them to read the user's preferences from the database and file, respectively.

UserSimilarity and itemSimilarity: UserSimilarity is used to defined the similarity between two users. It is a core part of the recommendation engine based on collaborative filtering and it can be used to calculate the user's "neighbors". Here, the current user's neighbors are the users with the same preferences. ItemSimilarity calculates the similarity between the contents.

UserNeighborhood: It is used for the recommended method based on user similarity. The recommended content is based on finding the neighbors with similar preferences to the current user. UserNeighborhood defined method to determine the neighbor users and concrete realization is generally calculated based on UserSimilarity.

Recommender: Recommender is the abstract interface of the recommendation engine and is the core component in the Taste. In the program, providing it with a DataModel, it can calculate the recommended content for different users. In practical application, it is mainly used as the realization of GenericUserBasedRecommender and GenericItemBasedRecommender, to achieve the recommendation engine based on user similarity and content-based, respectively.

CLOUD MINING REALIZATION BY USING TASTE ALGORITHM

Firstly, collect the video program information which the users liked to watch and mainly collect the data from the following several aspects:

- Video programs that users have viewed
- Some videos marked as favorite
- Long residence time of a video site
- User satisfaction rating of video programming. Through these actions, it could be determined what kinds of videos are preferred for the users

After extracting the user viewing behavior, the following two ways is used to combine the user's viewing behavior:

- Different behavior of grouping can be divided into "View", "collection" and so on and then calculate the different users/video similarity based on the different behavior. Which is similar to the Amazon given "collector of the book also purchased ...", "Viewer of the book also looked at ..." and so on
- According to reflects of different behavior, the degree of user preferences are weighted to obtain the overall preferences of the user for video. In general, the weight of explicit user feedback is larger than that of the implicit but it is rather sparse, after all, the explicit feedback from users is few. After collection of user ratings data, some pre-processing is required and the core of the work is: noise reduction. Noise reduction is to reduce the noise in the user viewing data. The noise is generated by users in the process of watching the video. And a lot of noise and user misuse may exist. The noise in the behavioral data can be filtered out by adopting classic data mining algorithms which could make our analysis more precise

Once having the user preferences video by the user viewing behavior analysis, the similar users and television programs could be obtained based on user preferences and the recommendation could be done based on the similar users or videos which are the most typical two branches of TasteCF: The user-based CF and item-based CF. These two methods have to compute similarity and several basic methods of calculating the similarity are based on the vector. Their key idea is to calculate the distance of two vectors and the closer distance the greater similarity.

In the recommended scenario and in two-dimensional matrix preferred by the user-video, a user preference of all video programming is formed as a vector to calculate the similarity between users, or put the preferences of all users of a video program as a vector to calculate the similarity between items. By using Euclidean distance similarity calculation, the calculation is described as follows.

Assume that x, y are two points of the n-dimensional space, the Euclidean distance between them is calculated as:

$$d(x,y)=\sqrt{\sum(x_i-y_i)^2} \quad (6)$$

Specially, when n = 2, Euclidean distance is the distance between two points in the plane.

When the similarity is expressed with the Euclidean distance, the following formula is generally used. Obviously, the smaller the distance is and the greater the similarity is:

$$Sim(x,y) = \frac{1}{1 + d(x, y)} \quad (7)$$

After calculating the similarity, the neighbor between users could be found according to the similarity. This is the Neighbor calculation method based on similarity threshold which is the maximum limit on the proximity of the neighbors. All points are considered as the neighbor of the current point which fall into the circular region with the current point as the center and the distance of K as the radius. The number of neighbors calculated in this method is not sure but the similarity does not appear large errors.

After the pre-calculation the adjacent users and adjacent videos could be got, based on which users could obtain the recommended videos.

The videos to users can be recommended by the most typical two branches of TasteCF: The user-based CF and item-based CF. First introduces the concept of two methods: The basic idea of CF based on the users is quite simple which is to find the adjacent neighbor users based on user preference items and then recommend to the current user like the neighbor users. In the calculation, a user preference for all items is used as a vector to calculate the similarity between users. After locating the k neighbors, according to neighbors of similar weight and their appetite for items, it can be predicted that the current user does not have the preferences which are not covered by items and then calculate a sorted list of items as recommended.

The principle of item-based CF is similar to that of user-based CF which uses the item itself to compute the neighbor, rather than from the user's perspective. The user-based CF is based on user preference for items to find similar items and then based on the user's historical preferences, recommend similar items. By analysis from the computational point of view which is to use all users preferences of an item as a vector to calculate the similarity of items and after getting similar items of items, according to the preferences of the user history, it predict that the current user has not expressed preference items and then it computes from a sorted list of items as recommended. Because the recommendation is video, it's not a social product and can not accurately fulfill certain similarity with you. And the close video to the users may be recommended. Because the person being recommended and those with the same viewing habits of users do not understand, if the given explanation is such people with the same interest as you, it's hard for users to be convinced. However, if the given recommendation reason is this video very similar to a video program you watched before, the user may think it is reasonable and to adopt this recommendation. For the above reasons, the CF based on items is adopted as the core algorithm video recommendation system.

The first step is to model the data, analyze the main entities involved in the application, including the relationships between entities and design database storage, classes in your program, as well as the DataModel of recommendation engines.

Figure 2 shows the structure of a user video entity-relationship model.

As shown in Fig. 2, the entities are defined as follows:

- **Video:** Represent a movie which contains the basic information about the video: number, name, date, type and so on
- **User:** Represent a user which contains the basic information for the user: number, name, e-mail and so on
- **Video reference:** Represent the preference degree of a user on a video which contains the user ID, Video ID, user ratings and a score of times

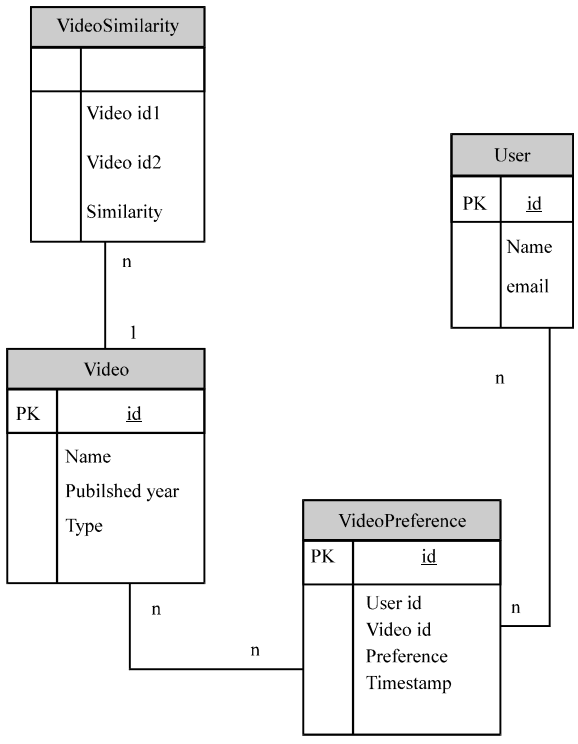

Fig. 2: Entity-relationship model of user video

- **Video similarity:** Represent the similarity of two videos which contains the similarity of two videos. The similarity between two videos can be getting from the basic information of the videos

Based on the above data model DateModel and the corresponding table in the database are created.

Implementing a recommendation engines needs to implement Recommender interface which is an extension of recommendation engine available for a certain Taste. It is an extension of GenericUserBasedRecommender. One of the most important ways is to instantiate the recommendation engine's constructor method and its construction method involves the following steps:

- In order to improve the speed of real-time response of the recommendation engine, the video information needs to be reprocessed. The similarity of the videos should be calculated well in advance and stored in the video_similarity table in the database and then they are read from the database and used to create a collection of ItemItemSimilarity
- Generate a content similarity ItemSimilarity based on the collection of ItemItemSimilarity
- Create an EmbededItemBasedRecommender instance which is an internal class containing a GenericItemBasedRecommender instance. In this recommend method, based on the video list of the users score in a DataModel, the most similar videos recommended to the users could be calculated by calling the method of mostSimilarItems in GenericItemBasedRecommender

CONCLUSION

The technologies of the digital video community and friends are not very mature. When using collaborative filtering algorithm based on the user, where the user may think that his "friends" simply do not know and hereby the videos may not be recommended to him by his "friend" watching a video However, by using the collaborative filtering algorithm based on content, current users watching a video will have a similar program recommended and users will find this recommendation is really their own interests. Consequently, application of collaborative filtering algorithm based on content is preferred in digital television recommended algorithm for cloud mining.

REFERENCES

Chaturvedi, A.D., P.E. Green and J.D. Caroll, 2001. K-modes clustering. J. Classification, 18: 35-56.

Gelbard, R., O. Goldman and I. Spiegler, 2007. Investigating diversity of clustering methods: An empirical comparison. Data Knowl. Eng., 63: 155-166.

Hongtao, X., Z. Qingsheng, Q. Hongchun and Y. Ruilong, 2011. User-based collaborative recommendation filtering algorithm using extremely valued ratings. JDCTA, AICIT, 5: 47-54.

Owen, S., R. Anil, T. Dunning and E. Friedman, 2011. Mahout in Action. Manning Publications Co., Greenwich, CT, USA., ISBN: 9781935182689, Pages: 387.

Sambasivam, S. and N. Theodosopoulos, 2006. Advanced data clustering methods of mining Web documents. Inform. Sci. Inform. Technol., 3: 564-579.

Tun, Z.W. and N. Thein, 2007. An approach of standardization and searching based on hierarchical bayesian clustering (HBC) for record linkage system. Proceedings of the 5th International Conference on Creating, Connecting and Collaborating through Computing, Volume 1, January 24-26, 2007, Kyoto, pp: 54-60.

Van Beek, P., J.R. Smith, T. Ebrahimi, T. Suzuki and J. Askelof, 2003. Metadata-driven multimedia access. IEEE Signal Process. Mag., 20: 40-52.

Wei, K., L. Nianlong and S. Zilei, 2011. Resource recommendation based on topic model for educational system. Proceedings of the 6th IEEE Joint International Information Technology and Artificial Intelligence Conference, Volume 2, August 20-22, 2011, Chongqing, pp: 370-374.