毕业设计论文

外文翻译

沈阳航空航天大学

外文翻译原文及译文

学　　院	计算机学院
专　　业	网络工程
班　　级	1734010402
学　　号	173401040213
姓　　名	姚茗瀚
指导教师	石祥滨
负责教师	

沈阳航空航天大学

2021 年 6 月

MVC Framework

Abstract. Association rule analysis algorithm is widely used in Web log analysis, but the existing association rule analysis algorithm will significantly reduce the analysis and mining performance when the amount of Web log is relatively large. This paper proposes an improved clustering algorithm, which first clusters users with the same interests and hobbies, and then mines association rules for users in the same category, thereby reducing data dispersion. Based on Django's MVC framework, it optimizes the storage and storage of Web logs. In the analysis part, users can configure the support and confidence of association rule mining through the front-end, and at the same time query the results of mining through Hive, and use encryption algorithms in the data transmission process to ensure data security.

Keywords. HDFS; Web log mining; clustering; FP-Growth algorithm

1. Introduction

The overall requirements of web log mining system design are safety, efficiency and ease of use. Use an optimized distributed file storage architecture to save log data, and use encryption algorithms to ensure data security during log transmission. At the same time, it uses distributed computing tools to extract useful features in Web log data, combines with improved clustering algorithm to classify the log data, and then finds the rule sequence that meets the attribute requirements through the association rule mining algorithm. Users can manage the tasks created by the system, set the parameters of the association rule mining algorithm and obtain the results of the mining tasks in time.

2. Optimization of K-means clustering algorithm

For the mining of association rules of massive Web log data, the main algorithms used are Apriori and FP-Growth algorithms. They all have their own advantages and disadvantages in the execution efficiency of the algorithm [1]. The Apriori algorithm requires constant access to the database, and the overhead of the database is obviously unacceptable. The FP-Growth algorithm needs to store it in memory when constructing FP-Tree, but the FP-Tree constructed by massive Web log data will consume most of the memory, which will seriously affect the performance of the cluster [2]. Therefore, this paper proposes an FP-Growth association rule mining algorithm based on the improved K-means clustering algorithm. First, the proposed clustering optimization algorithm is used to reduce data dispersion, and users with the same hobbies are classified into one category, and then the same Mining association rules for class data [3].

2.1. Analysis of K-means clustering algorithm

Compared with other clustering algorithms, K-means algorithm has the advantages of simple execution process and fast convergence speed and is easy to implement. To measure the performance of the K-means clustering algorithm, it is often explained by the sum of the square error (SSE). The specific calculation method of the sum of the square error is shown in formula (1).

$$SSSSSS = \sum^{kk}{}_{ii=1} \sum_{xx \in ccii} dddddddd(xx, cc_{ii}) \qquad (1)$$

Where CC_{ii} represents the particle in the i-th cluster, x represents any data point in the i-th cluster, so the formula $\sum_{xx \in ccii} dddddddd(xx,\ cc_{ii})$ represents the sum of the distances from all the data points in the i-th cluster to the particle points of the cluster, and the K-means The distance calculation method uses the Euclidean distance calculation method. Therefore, SSE represents the sum of the distances between all data points and the mass point of the cluster to which the point belongs. If the SSE value is larger, it means that the clustering effect of each cluster is not very good, and the data points are not very dense; If the value is smaller, it means that the clustering effect between each cluster is better. The calculation here can only start when the initial mass points are determined, so it is only a local optimal solution, because the K-means algorithm does not have a clear method to confirm the initial mass points. If the initial mass points are not well selected, it will cause SSE is too large. Kim et al. determined the initial value of K according to the idea of maximum and minimum distance. First, calculate the set of minimum distance between each point and each cluster point, and then select the point with the largest distance as the new cluster point. This can avoid the clustering effect being too close due to the selection of cluster points, but this method can not solve the influence of abnormal points and the consumption of iterative calculation of new cluster points.

2.2. Optimization of K-means algorithm

Based on the analysis of the K-means algorithm, in view of the shortcomings of the K-means algorithm and combined with the data characteristics of the Web log itself, this chapter proposes some improved solutions, mainly including the following:

- Web log data preprocessing. Among the massive Web log data, not all data records belong to normal user access. Some of the abnormal data can be eliminated according to the request status code is not in the normal range, the request method is not GET, and the requested resource type is not a page request. Filtering out these logs that do not meet the request and the status code error can reduce the number of abnormal points in the Web log data, which can prevent some of the extreme attribute data from having a serious impact on the calculation of the sample distance.□□
- Optimization of the number of initial clusters. The selection of the initial clustering center will seriously affect the final clustering effect, so random selection of k clustering centers is not ideal. The Web log analysis in this article is based on a big data platform, so the number of clusters can be determined based on distributed computing. First of all, because the calculation method of SSE only considers the local optimality, it does not consider the difference of the particles in each cluster. Therefore, this chapter proposes the calculation method of the global optimal solution to determine the optimal initial cluster number. The specific definition of the function is as follows:□

ii

$$OOOOOO(KK) = \frac{\sum_{i=1}^{k}\sum_{x\in c} sim(x,c_i)}{\sum ii=1,jj=1ssiiss(ccii,ccjj)} {}_{kk}(nn-kk) \qquad (2)$$

kk

- In formula (2), k represents the number of clusters, n represents the number of all data, and sim represents the distance between two data points. The specific calculation method of the distance will be explained in the next section. The formula $\sum^{kk}{}_{ii=1,jj=1}ddddss(cc_{ii},\ cc_{jj})$ represents the sum of square errors of the mass points between each cluster. The larger the value, the farther
the distance between each cluster, the more obvious the data aggregation, and the better the clustering effect. The formula represents the sum of squared errors within the group, which represents the convergence of each cluster. The smaller the value, the better the clustering effect in each cluster. Therefore, based on the algorithm of the global loss function, in accordance with the definition of the global loss function, this paper determines the number of cluster families by finding the k value with the largest fluctuation. The value of the largest fluctuation can be determined by finding the turning point at which the rate of change suddenly becomes larger, because if the rate of change tends to be flat, it means that it is meaningless to continue to increase the number of clusters. When each point when it is a cluster, the global loss function is zero.□□
  - Iterative process optimization. In the K-means algorithm, the clustering point for the next iteration is determined by the mean point of all the data in the cluster. The cluster center formed in this way is probably not in the real high-density area of the data, which leads to the final clustering. The class results will have a certain deviation, and the iteration cost will become higher. This paper proposes an optimized binary clustering algorithm based on the binary K-means algorithm combined with the maximum distance idea. First find out the cluster with the largest square error sum in the group, calculate the K points with the largest distance from the cluster particles, and perform the binary clustering of the cluster according to these K points, and then obtain K binary cluster sets, choose The square error and the smallest division

replace the original cluster. Keep repeating the above steps until the number of clusters obtained is equal to the initial set K.□□

3. Web log storage and analysis system interface design

As a complete Web log mining system, the system mainly implements the following interfaces in function:

- User interface, which is mainly used to verify the legitimacy of users. It is mainly divided into ordinary users and administrator users. Users of different levels have different permissions;
- Log storage interface, this interface is mainly used for users to upload Web log data that needs to be analyzed. Through this interface, the back-end optimized HDFS storage architecture can be triggered to save data;
- Data download interface, which is mainly used to download Web log data and the execution results of mining tasks, including classified data and corresponding association rule mining results;
- Task creation interface, this interface is used to create mining tasks, the user can select data batches and set the parameters of the association rule mining algorithm through this interface;
- Status query interface, through which users can view the execution status of tasks. If the task execution fails, you can view the failure log, and then restart the task; if the execution is successful, you can download the mining results.

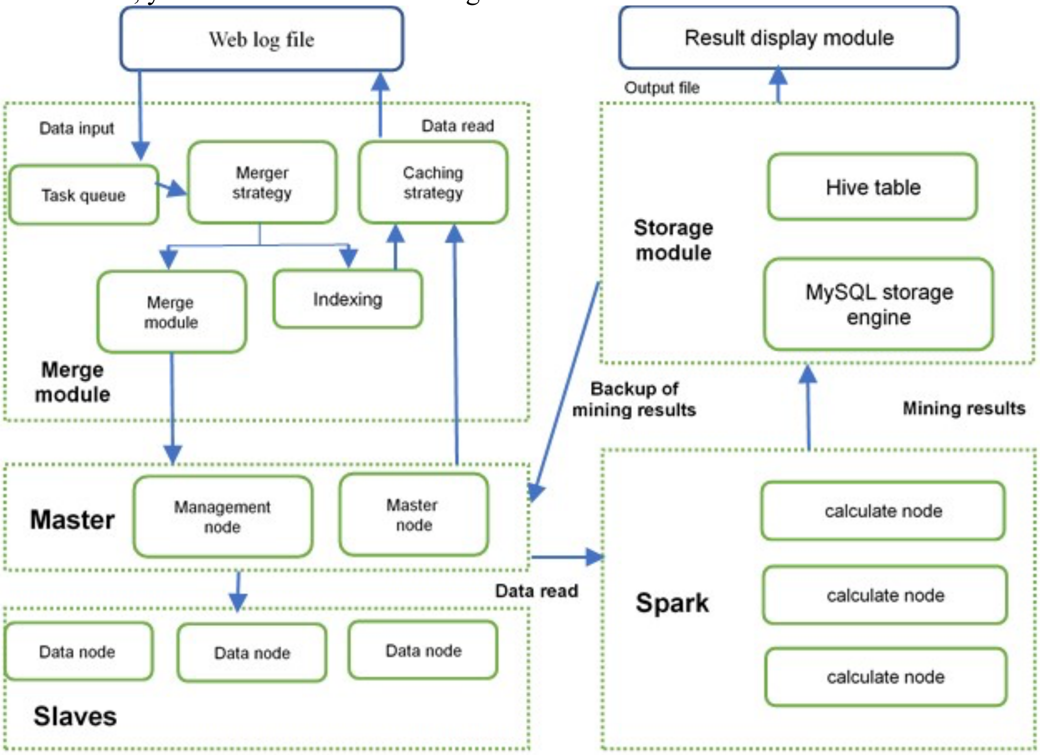

Figure 1. System overall design architecture diagram

-

4. System framework design

The import of Web logs is mainly based on the HTTP protocol. Through the configuration of the client, select the Web log data to be imported [4]. The uploaded Web logs are stored in the HDFS after the file merging module, as the basis of data analysis. Then, based on the Spark cluster, mining the Web logs. Web log mining mainly includes cluster analysis and association rule mining. The results of log mining are stored in MySQL and Hive tables, and are persisted to HDFS at the same time. Users can obtain mining information through the result display module. Figure 1 shows the overall system design architecture diagram.

4.1. System function design

Figure 2 shows the functional structure diagram of the Web log mining system. The system is mainly divided into three modules: The log management module is mainly responsible for the upload and download functions of logs; the log mining module is mainly responsible for user management tasks, viewing task status and querying task execution results; the user management module is mainly responsible for managing the user's login registration information.□

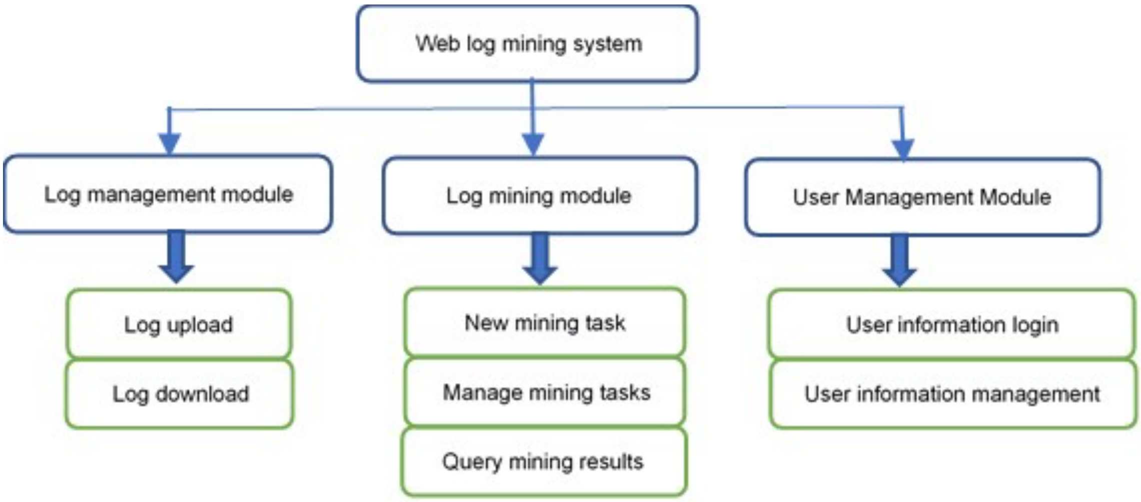

Figure 2. System function structure diagram

4.2. Database Design

When users use the system to perform mining tasks, in addition to storing Web logs in HDFS, they also need to store some user and task-related tables in the MySQL database to show users related information about task execution and data mining the result of. At the same time, data block tables are also needed to store user-related information, which mainly involves the following four tables:

- User information table. As shown in Table 1, it is a user information table, which is mainly provided to users of the super user management system. The field user_role is the user's role, the value is admin for super user, and the value is engineer for ordinary user. Super users can modify common user information by adding, deleting, modifying and checking. **Table 1.** User information table

Field Name	Types	Field constraint	Field description
user_id	INTEGER	Primary key	User ID
user_name	VARCHAR[64]	non empty	user name
user_role	VARCHAR[64]	non empty	User role
Creat_time	DATETIME	non empty	Registration time
remarks	VARCHAR[64]	naught	Remarks

- File storage table

  When the Web log is imported into HDFS, it is necessary to store the log file information in MySQL according to the batch number of the file. Among them, the information of the log file in Mysql is consistent with the information of the HDFS file. As shown in Table 2, the main fields included in the log storage information, the pat field indicates the location of the Web log file on the HDFS, and the batch ID is the unique representation of each batch of logs.

Table 2. File storage table

Field Name	Types	Field constraint	Field description
batch_id	INTEGER	Primary key	Batch ID
batch_name	VARCHAR[64]	non empty	Batch name
path	VARCHAR[64]	non empty	Storage path
Creat_time	DATETIME	non empty	Storage time

- Task execution information table

Table 3 shows the main fields and descriptions of the task execution information table. The table mainly stores the information of users performing mining tasks. Among them, user_id and batch_id are respectively associated with user information and file storage. Task_name represents the name of the task execution, which is mainly composed of the timestamp and the uploaded folder name. The status field indicates the execution status of the task (0: ready, 1: executing, 2: executing successfully, 3: executing failed).The conf_info field indicates that the user's execution task is the selected configuration information, that is, the support and confidence when mining log association rules.

Table 3. Task execution information table

Field Name	Types	Field constraint	Field description
task_id	INTEGER	Primary key	Batch ID
user_id	INTEGER	Foreign key	User ID
batch_id	INTEGER	Foreign key	Log batch ID
task_name	VARCHAR[64]	non empty	mission name
creat_time	DATETIME	non empty	Storage time
start_time	DATETIME	non empty	Starting time
finish_time	DATETIME	non empty	End Time
status	INTEGER	non empty	Execution status
conf_info	VARCHAR[64]	non empty	Configuration information

- Result information table

As shown in Table 4, it is a table of related information stored in the execution result after the task is successfully executed. This table mainly stores the result information of Web log clustering mining and association rule mining. Among them, the field result_id represents the unique identifier of each mining result, and the task_id is the foreign key that associates the specific information of each mining task. Cluster_nuni represents the number of clusters in this batch of Web journals mined by the improved K-means algorithm, and the log information of each cluster is stored on the HDFS, and the specific path information is stored in the field cluster_path. The freeq_num table 7K is the total number of frequent items mined by the cluster-based FP-Growth algorithm, and the information of the specific frequent items is stored on the HDFS through the path field freeq_path. fp_growth_num represents the total number of association rule mining of FP_Growth conversion method, that is, after combining with the improved K-means clustering algorithm, the total number of association rules mined by the association rule mining algorithm for each type of users with similar interests, and The specific association rule information is also stored on the HDFS through the path fp_growth_path. At the same time, the data about Web log information and execution result information stored in HDFS will be imported into the Hive table, which makes it convenient for users to query the results through the Hive table and display the corresponding execution results to the user through the front-end page.

Table 4. Result information table

Field Name	**Types**	**Field constraint**	**Field description**
result_id	INTEGER	Primary key	Result ID
task_id	INTEGER	Foreign key	Batch ID
user_id	INTEGER	Foreign key	User ID
batch_id	INTEGER	Foreign key	Log batch ID
task_name	VARCHAR[64]	non empty	mission name
creat_time	DATETIME	non empty	Storage time
Cluster_num	INTEGER	non empty	Number of clusters
Cluster_path	VARCHAR[256]	non empty	Cluster storage path
Freq_num	INTEGER	non empty	Number of frequent items
Freq_path	VARCHAR[256]	non empty	Frequent item path
Fp_growth_num	INTEGER	non empty	Total number of association rules
Fp_growth_path	VARCHAR[256]	non empty	Rule storage path
conf_info	VARCHAR[256]	non empty	Configuration information

5. System implementation

5.1. MVC framework construction

Django is an open source Web application framework written by Python. Figure 3 shows the overall structure of Django. The code management of the entire system framework of Django is mainly composed of the following files:

- Urls.py: This file is used to receive the user's request to access API, and then jump to the view according to the user's request. The corresponding interface in py. As shown in Table 5-5, the mapping relationship between the system url and the interface is defined, for example, "url(r'Alogin', views.login, name='login')", the user accesses the system through the HTML protocol. Page, it will pass urls. The py file requests the user login interface in the view, and then returns the login result. In addition to the user login interface, it mainly includes the interface for uploading and downloading logs, the interface for creating tasks, and the interface for requesting task execution result information.
- Views.py: User-defined interface, that is, receive urls. The user request forwarded by py, and then the specific implementation logic of each request is defined in this file. As shown in Table 5-6, it is the service interface for users to request execution result information. First, obtain the server address of the user's download request, then search for the corresponding task_id, and then obtain the specific execution result information according to the task information.
- Models.py: Related to database operations, when users request task status and corresponding execution result information, they need to connect to the database and then get specific data. As shown in Table 5-7, model. The class definition for obtaining task execution information in py is mainly to connect to the database, and then connect each field in the database with the previous

The fields displayed in the table on the end correspond to the specific logic in view. Realized in py.

- Admin.py: Complete the background configuration by adding configuration code.
- Settings.py: Store the configuration information of Djaango, such as the location of static files, etc.

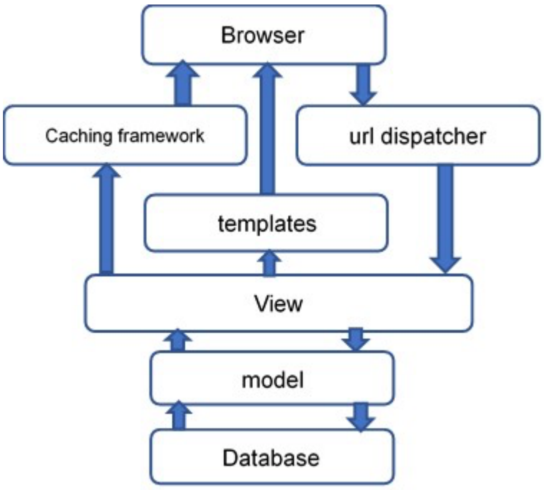

Figure 3. Django structure diagram

5.2. System function module realization For web log collection, real-time message systems are generally used to collect, such as kafta and nsq message queue [5]. According to the different business scenarios of the website, different topics are used to collect different Web logs, and then the log files are saved to disk through HDFS. The mining system designed in this paper mainly refers to the mining of the association rules of offline Web logs. The analyzed Web logs are based on the log data left after the user visits the website, rather than processing the logs being generated in real time. Therefore, to mine and analyze the logs, the user needs to select the location of the imported log data, and then start uploading the Web log.□□

In order to prevent the web log data uploaded by users from being tampered with, it is necessary to strengthen the security of the web log during the upload process. First, select the encryption algorithm. Commonly used encryption algorithms include symmetric encryption and asymmetric encryption algorithms. However, the sub-asymmetric encryption algorithm needs to use the public key and the secret key, and the encryption and decryption process takes a long time. Therefore, the AES symmetric encryption algorithm is selected to encrypt the uploaded Web log. However, because the secret keys used in the encryption and decryption process of the symmetric encryption algorithm are the same, the security is relatively low. In order to strengthen security, MD5 of the log is used as a secondary check [6]. That is to calculate MD5 for the encrypted log, and then use the comma as the separator to connect the encrypted string to form a new string. In order to reduce the bandwidth consumption in the log transmission process, the new string is compressed through the gzip compression algorithm and then uploaded to the server through the HTTP protocol.□□

When the server receives the message, it first decompresses gzip, and then obtains the encrypted string and the corresponding MD5 value through the separator. Then calculate the MD5 value. If the MD5 value calculated for the encrypted string is the same as the transmitted MD5 value, it means that the transmitted data has not been modified; if the MD5 value is not the same, it means that the data has been modified during transmission, and then the data is discarded. After the MD5 value is verified, the encrypted string is decrypted with the same AES key, and the decoded string is the uploaded Web log. After the backend obtains the decrypted Web log, it inputs the log into the file pre-processing module [7].

After receiving the decrypted log data, the data is pre-processed. Data pre-processing is a necessary process in Web log mining and the core work of the entire data preparation [8]. Data pre-processing is the foundation of the entire mining process. If the data is not pre-processed well, it will directly affect the rules and patterns generated in the mining process, and it is also a guarantee of mining quality. Data pre-processing mainly includes the stages of data cleaning, user identification, session identification and path supplementation [9].

- Data cleaning

  In the original Web log, there are many requests with status codes of 3XX series and 4XX series [10]. These requests indicate redirection or request errors, and also include some requests for web resources with suffixes such as gif and jpg. It is meaningless to analyze user behaviour, so it needs to be filtered out from the original data, and only the GET request with the status code of 2XX series needs to be retained.
- User identification

  The user identification stage divides the visits of different users from the data after data cleaning, that is, the user IP is the key and the value is the user's access item, and each access item is composed of the access link and the access time [11].
- Session recognition and path supplement

  Session recognition refers to identifying a complete browsing process of a user, that is, a series of page sequence collections visited by the user from visiting the site to leaving the site. This is called a session of the user. The system sets a time threshold of 30 minutes for each session, that is, the time of a session will not exceed the threshold [12]. Due to the impact of the cache of the website proxy server, the user's access request will not generate the corresponding log, so it is necessary to add the access request missed by these servers to the user session to provide a complete data source for Web log mining.

6. Conclusion

This paper designs a visual Web log mining system based on big data platform. On the basis of Django's MVC framework, with the help of the open source Bootstrap framework, a Web log storage and mining system for users is realized. This chapter details the internal implementation details of each module function, and shows the overall framework of the language system and the specific information of each module. Using this system, users can realize the Web log storage, cluster analysis and association rule mining proposed in this paper through simple front-end operations.

Acknowledgments

Thanks for the following fund for supporting this article: Research Project of Lanzhou Vocational and Technical College “Research on the Construction of Innovation and Entrepreneurship Teams in Higher Vocational Colleges Based on the Integration of Production and Education under the ‘Internet +’ Perspective”.

MVC 架构

摘要。关联规则分析算法在 Web 日志分析中得到了广泛的应用，但现有的关联规则分析算法在 Web 日志量比较大的情况下会明显降低分析和挖掘性能。本文提出了一种改进的聚类算法，首先对具有相同兴趣和爱好的用户进行聚类，然后对同一类别的用户进行关联规则挖掘，从而减少数据的分散性。基于 Django 的 MVC 框架，它优化了网络日志的存储和储存。在分析部分，用户可以通过前端配置关联规则挖掘的支持度和置信度，同时通过 Hive 查询挖掘结果，并在数据传输过程中使用加密算法，保证数据安全。

Keywords.HDFS；网络日志挖掘；聚类；FP-Growth 算法

1.绪论

网络日志挖掘系统设计的总体要求是安全、高效和易于使用。使用优化的分布式文件存储架构来保存日志数据，并使用加密算法来保证日志传输过程中的数据安全。同时，利用分布式计算工具提取 Web 日志数据中的有用特征，结合改进的聚类算法对日志数据进行分类，再通过关联规则挖掘算法找到符合属性要求的规则序列。用户可以对系统创建的任务进行管理，设置关联规则挖掘算法的参数，及时获得挖掘任务的结果。

2.K-means 聚类算法的优化

对于海量网络日志数据的关联规则的挖掘，主要使用的算法是 Apriori 和 FP-Growth 算法。它们在算法的执行效率上都有各自的优势和劣势[1]。Apriori 算法需要不断访问数据库，数据库的开销显然是不可接受的。FP-Growth 算法在构建 FP-Tree 时需要存储在内存中，但由海量 Web 日志数据构建的 FP-Tree 会消耗大部分的内存，这将严重影响集群的性能[2]。因此，本文提出一种基于改进的 K-means 聚类算法的 FP-Growth 关联规则挖掘算法。首先，利用提出的聚类优化算法，减少数据的分散性，将具有相同爱好的用户归为一类，然后对类数据进行相同的挖掘关联规则[3]。

2.1.K-means 聚类算法的分析

与其他聚类算法相比，K-means 算法具有执行过程简单、收敛速度快、易于实现的优点。为了衡量 K-means 聚类算法的性能，通常用平方误差之和（SSE）来解释。平方误差之和的具体计算方法如公式（1）所示。

$$SSSSSS = \sum^{kk}{}_{i=1}\sum_{x\in cci} ddddddd(x, cc_i) \quad （1）$$

其中，C_i代表第 i 个簇中的粒子，x 代表第 i 个簇中的任何数据点，所以公式$\sum_{x\in cci} ddd$（x。cc_i）表示第 i 个聚类中的所有数据点到聚类中的粒子点的距离之和，K-means 距离计算方法采用欧氏距离计算方法。因此，SSE 代表所有数据点与该点所属簇的质量点之间的距离之和。如果 SSE 值较大，说明各聚类的聚类效果不是很好，数据点不是很密集；如果该值较小，说明各聚类之间的聚类效果较好。这里的计算只有在初始质量点确定后才能开始，所以它只是一个局部最优解，因为 K-means 算法没有明确的方法来确认初始质量点。如果初始质量点选得不好，会造成 SSE 过大。Kim 等人根据最大和最小距离的思想来确定 K 的初始值。首先，计算每个点与每个群集

点之间的最小距离集合，然后选择距离最大的点作为新的群集点。这种方法可以避免因聚类点的选择而导致聚类效果过于接近，但这种方法不能解决异常点的影响和新聚类点迭代计算的消耗问题。

2.2.K-means 算法的优化

在分析 K-means 算法的基础上，针对 K-means 算法的缺陷，结合网络日志本身的数据特点，本章提出了一些改进的解决方案，主要包括以下内容。

- 网络日志数据的预处理。在大量的 Web 日志数据中，并非所有的数据记录都属于正常的用户访问。根据请求状态码不在正常范围内、请求方式不是 GET、请求资源类型不是页面请求等情况，可以剔除部分异常数据。过滤掉这些不符合请求和状态码错误的日志，可以减少 Web 日志数据中的异常点，可以避免一些极端属性数据对样本距离的计算产生严重影响。
- 优化初始聚类的数量。初始聚类中心的选择会严重影响最终的聚类效果，所以随机选择 k 个聚类中心并不理想。本文的 Web 日志分析是基于大数据平台的，所以聚类的数量可以基于分布式计算来确定。首先，由于 SSE 的计算方法只考虑了局部最优性，没有考虑每个簇中粒子的差异。因此，本章提出了全局最优解的计算方法来确定最优初始簇数。该函数的具体定义如下。

$$OPT(K)=\frac{\sum_{i=1}^{k}\sum_{x\in c_i} sim(x,c_i) \Big/ _{(n-k)}}{\sum_{i=1,j=1}^{k} sim(c_i,c_j) \Big/ _{k}}$$

•

i

在公式（2）中，k 代表聚类的数量，n 代表所有数据的数量，sim 代表两个数据点之间的距离。距离的具体计算方法将在下一节解释。公式中$\sum^{kk}{}_{i=1,jj1ddddss(}cc_i, cj)$表示每个簇之间质量点的平方误差之和。该值越大，就越远

每个聚类之间的距离越大，数据聚集越明显，聚类效果越好。该公式表示组内的平方误差之和，代表每个聚类的收敛程度。该值越小，每个簇的聚类效果越好。因此，在全局损失函数的算法基础上，按照全局损失函数的定义，本文通过寻找波动最大的 k 值来确定聚类族的数量。最大波动的值可以通过寻找变化率突然变大的转折点来确定，因为如果变化率趋于平缓，就意味着继续增加聚类的数量是没有意义的。当每个点当它是一个集群时，全局损失函数为零。

- 迭代过程的优化。在 K-means 算法中，下一次迭代的聚类点是由聚类中所有数据的平均点决定的。这样形成的聚类中心很可能不在数据真正的高密度区域，这就导致了最终的聚类。类结果会有一定的偏差，迭代成本也会变高。本文在二元 K-means 算法的基础上，结合最大距离思想，提出一种优化的二元聚类算法。首先找出群中平方误差总和最大的聚类，计算出与聚类粒子距离最大的 K 点，并根据这 K 点对聚类进行二元聚类，然后得到 K 个二元聚类集，选择 平方误差和最小的划分取代原有聚类。不断重复上述步骤，直到获得的聚类数量等于初始集 K。

3.网络日志存储和分析系统界面设计

作为一个完整的网络日志挖掘系统，该系统在功能上主要实现了以下接口。

- 用户界面，主要用于验证用户的合法性。它主要分为普通用户和管理员用户。不同级别的用户有不同的权限。
- 日志存储接口，该接口主要用于用户上传需要分析的 Web 日志数据。通过这个接口，可以触发后端优化的 HDFS 存储架构来保存数据。
- 数据下载界面，主要用于下载 Web 日志数据和挖掘任务的执行结果，包括分类数据和相应的关联规则挖掘结果。
- 任务创建界面，该界面用于创建挖掘任务，用户可以通过该界面选择数据批次，设置关联规则挖掘算法的参数。
- 状态查询界面，用户可以通过该界面查看任务的执行状态。如果任务执行失败，可以查看失败日志，然后重新启动任务；如果执行成功，可以下载开采结果。

4.系统框架设计

网络日志的导入主要是基于 HTTP 协议的。通过客户端的配置，选择需要导入的 Web 日志数据[4]。上传的 Web 日志经过文件合并模块存储在 HDFS 中，作为数据分析的基础。然后，在 Spark 集群的基础上，对 Web 日志进行挖掘。网络日志挖掘主要包括聚类分析和关联规则挖掘。日志挖掘的结果存储在 MySQL 和 Hive 表中，并同时持久化到 HDFS。用户可以通过结果显示模块获得挖掘信息。图 1 是系统整体设计架构图。

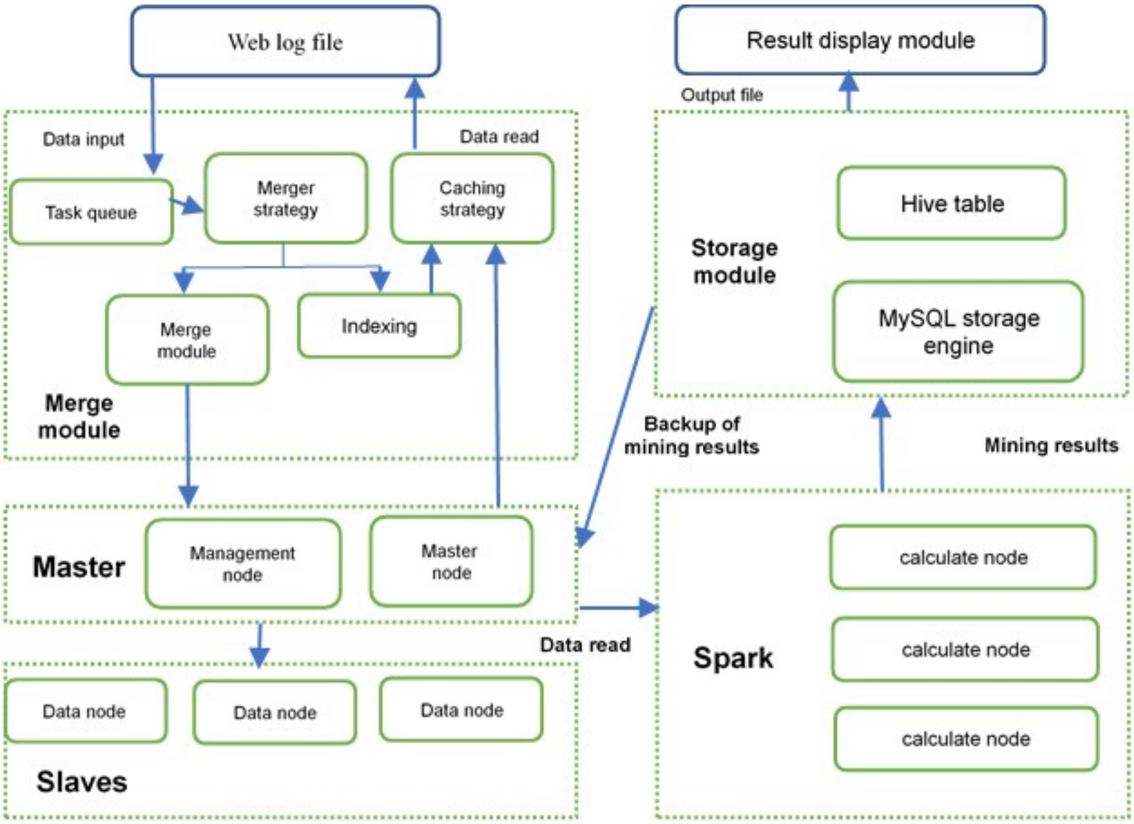

图 1.系统整体设计架构图

4.1.系统功能设计

图 2 是网络日志挖掘系统的功能结构图。该系统主要分为三个模块。日志管理模块主要负责日志的上传和下载功能；日志挖掘模块主要负责用户管理任务、查看任务状态和查询任务执行结果；用户管理模块主要负责管理用户的登录注册信息。

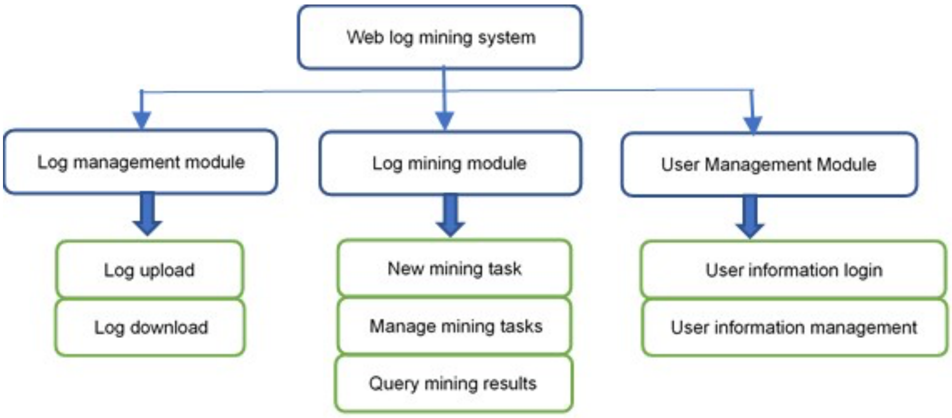

图 2.系统功能结构图

4.2.数据库设计

当用户使用系统进行挖掘任务时，除了在 HDFS 中存储 Web 日志外，还需要在 MySQL 数据库中存储一些用户和任务相关的表，以显示用户有关任务执行和数据挖掘结果的相关信息。同时，还需要数据块表来存储用户相关信息，主要涉及以下四个表。

- 用户信息表。如表 1 所示，它是一个用户信息表，主要提供给超级用户管理系统的用户。字段 user_role 是用户的角色，超级用户的值是 admin，普通用户的值是工程师。超级用户可以通过添加、删除、修改和检查来修改普通用户信息。**表 1.**用户信息表

领域名称	类型	领域约束	领域描述
user_id	INTEGER	主键	用户 ID
用户名称	VARCHAR[64]	非空	用户名称
用户角色	VARCHAR[64]	非空	用户角色
创建时间	日期	非空	注册时间
发言	VARCHAR[64]	无	备注

- 文件存储表

当 Web 日志被导入 HDFS 时，需要根据文件的批号将日志文件的信息存储在 MySQL 中。其中，Mysql 中的日志文件信息与 HDFS 文件的信息是一致的。如表 2 所示，日志存储信息中包括的主要字段，pat 字段表示网络日志文件在 HDFS 上的位置，批号是每批日志的唯一表示。

表 2. 文件存储表

领域名称	类型	领域约束	领域描述
batch_id	INTEGER	主键	批量标识
batch_name	VARCHAR[64]	非空	批次名称
路	VARCHAR[64]	非空	存储路径
创建时间	日期	非空	储存时间

- 任务执行信息表

表 3 显示了任务执行信息表的主要字段和描述。该表主要存储执行采矿任务的用户信息。其中，user_id 和 batch_id 分别与用户信息和文件存储相关。Task_name 表示任务执行的名称，主要由时间戳和上传的文件夹名称组成。status 字段表示任务的执行状态（0：准备就绪，1：

正在执行，2：执行成功，3：执行失败）。conf_info 字段表示用户执行任务时选择的配置信息，即挖掘日志关联规则时的支持度和置信度。

表 3.任务执行信息表

领域名称	**类型**	**领域约束**	**领域描述**
Task_id	INTEGER	主键	批量标识
user_id	INTEGER	外键	用户 ID
batch_id	INTEGER	外键	日志批号
任务名称	VARCHAR[64]	非空	任务名称
创造_时间	日期	非空	储存时间
开始时间	日期	非空	开始时间
结束时间	日期	非空	结束时间
身份	INTEGER	非空	执行状态
conf_info	VARCHAR[64]	非空	配置信息

- 结果信息表

如表 4 所示，它是任务成功执行后存储在执行结果中的相关信息表。该表主要存储 Web 日志聚类挖掘和关联规则挖掘的结果信息。其中，字段 result_id 代表每个挖掘结果的唯一标识符，task_id 为外键，关联每个挖掘任务的具体信息。Cluster_nuni 表示这批通过改进的 K-means 算法挖掘的网络期刊的集群数量，每个集群的日志信息存储在 HDFS 上，具体路径信息存储在字段 cluster_path 中。freeq_num 表 7K 是基于集群的 FP-Growth 算法挖掘出的频繁项总数，具体频繁项的信息通过路径字段 freeq_path 存储在 HDFS 上。fp_growth_num 表示 FP_Growth 转换方法的关联规则挖掘总数，即与改进的 K-means 聚类算法相结合后，针对每一类兴趣相近的用户，由关联规则挖掘算法挖掘的关联规则总数，具体的关联规则信息也通过路径 fp_growth_path 存储在 HDFS 上。同时，将存储在 HDFS 中的关于 Web 日志信息和执行结果信息的数据导入到 Hive 表中，方便用户通过 Hive 表查询结果，并通过前端页面向用户显示相应的执行结果。

表 4.结果信息表

领域名称	**类型**	**领域约束**	**领域描述**
result_id	INTEGER	主键	结果 ID
Task_id	INTEGER	外键	批量标识
user_id	INTEGER	外键	用户 ID
batch_id	INTEGER	外键	日志批号
任务名称	VARCHAR[64]	非空	任务名称
创造_时间	日期	非空	储存时间
集群_num	INTEGER	非空	集群的数量
集群_路径	VARCHAR[256]	非空	集群存储路径
频率_num	INTEGER	非空	经常性项目的数量
频率路径	VARCHAR[256]	非空	频繁出现的项目路径
增长数	INTEGER	非空	关联规则总数

Fp_growth_path	VARCHAR[256]	非空	规则存储路径
conf_info	VARCHAR[256]	非空	配置信息

5.系统实施

5.1.MVC 框架的构建

Django 是一个由 Python 编写的开源 Web 应用框架。图 3 显示了 Django 的整体结构。Django 的整个系统框架的代码管理主要由以下文件组成。

- Urls.py。这个文件用来接收用户访问 API 的请求，然后根据用户的请求跳转到视图。py 中的相应接口。如表 5-5 所示，定义了系统 url 和接口之间的映射关系，例如，"url(r'Alogin', views.login, name='login')"，用户通过 HTML 协议访问系统。页面，它将传递 urls。py 文件请求视图中的用户登录界面，然后返回登录结果。除了用户登录界面外，主要包括上传和下载日志的界面，创建任务的界面，以及请求任务执行结果信息的界面。
- View.py:用户定义的接口，也就是接收 urls。py 转发的用户请求，然后每个请求的具体执行逻辑都在这个文件中定义。如表 5-6 所示，它是用户请求执行结果信息的服务接口。首先获得用户下载请求的服务器地址，然后搜索相应的 task_id，再根据任务信息获得具体的执行结果信息。
- Models.py。与数据库操作有关，当用户请求任务状态和相应的执行结果信息时，需要连接到数据库，然后获得具体的数据。如表 5-7 所示，模型。py 中获取任务执行信息的类定义主要是连接到数据库，然后将数据库中的每个字段与之前的
  在末端的表格中显示的字段与视图中的具体逻辑相对应。在 py 中实现的。
- Admin.py。通过添加配置代码完成后台配置。
- Settings.py:存储 Djaango 的配置信息，如静态文件的位置等。

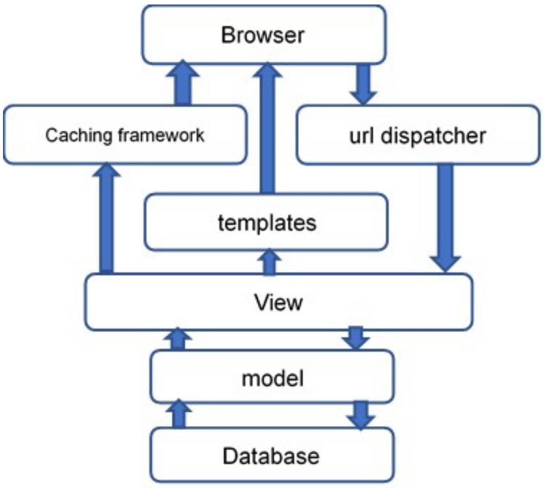

图 3. Django 结构图

5.2.系统功能模块实现 对于 Web 日志的采集，一般采用实时消息系统进行采集，如 kafta 和 nsq 消息队列[5]。根据网站的不同业务场景，采用不同的主题来采集不同的 Web 日志，然后通过 HDFS 将日志文件保存到磁盘。本文所设计的挖掘系统主要是指对离线 Web 日志的关联规则进行挖掘。所分析的 Web 日志是基于用户访问网站后留下的日志数据，而不是实时处理正在产生的日志。因此，为了挖掘和分析日志，用户需要选择导入日志数据的位置，然后开始上传 Web 日志。

为了防止用户上传的网络日志数据被篡改，有必要在上传过程中加强网络日志的安全性。首先，选择加密算法。常用的加密算法包括对称加密算法和非对称加密算法。但是，次不对称加密算法需要使用公钥和秘钥，加解密过程需要很长的时间。因此，选择 AES 对称加密算法对上传的网络日志进行加密。但是，由于对称加密算法的加密和解密过程中使用的秘钥是相同的，所以

安全性相对较低。为了加强安全性，日志的 MD5 被用来作为辅助检查[6]。即对加密后的日志计算 MD5，然后用逗号作为分隔符连接加密后的字符串，形成一个新的字符串。为了减少日志传输过程中的带宽消耗，新字符串通过 gzip 压缩算法进行压缩，然后通过 HTTP 协议上传到服务器。

当服务器收到信息时，它首先解压 gzip，然后通过分离器获得加密的字符串和相应的 MD5 值。然后计算出 MD5 值。如果计算出的加密字符串的 MD5 值与传输的 MD5 值相同，说明传输的数据没有被修改过；如果 MD5 值不相同，说明数据在传输过程中被修改过，然后丢弃该数据。在验证了 MD5 值后，用相同的 AES 密钥对加密字符串进行解密，解密后的字符串就是上传的网络日志。后台获得解密的网络日志后，将日志输入到文件预处理模块[7]。

收到解密后的日志数据后，要对数据进行预处理。数据预处理是网络日志挖掘的一个必要过程，也是整个数据准备的核心工作[8]。数据预处理是整个挖掘过程的基础。如果数据预处理不好，将直接影响挖掘过程中产生的规则和模式，也是挖掘质量的保证。数据预处理主要包括数据清洗、用户识别、会话识别和路径补充等阶段[9]。

- 数据清理

  在原始网络日志中，有许多状态代码为 3XX 系列和 4XX 系列的请求[10]。这些请求表示重定向或请求错误，还包括一些后缀为 gif 和 jpg 的网络资源请求。这对于分析用户行为是没有意义的，所以需要从原始数据中过滤掉，只需要保留状态码为 2XX 系列的 GET 请求。

- 用户识别

  用户识别阶段从数据清洗后的数据中划分出不同用户的访问，即以用户 IP 为关键，以用户的访问项目为值，每个访问项目由访问链接和访问时间组成[11]。

- 会话识别和路径补充

  会话识别是指识别用户的一个完整的浏览过程，即用户从访问网站到离开网站所访问的一系列的页面序列集合。这被称为用户的一个会话。系统为每个会话设置了 30 分钟的时间阈值，也就是说，一个会话的时间不会超过这个阈值[12]。由于受到网站代理服务器缓存的影响，用户的访问请求不会产生相应的日志，因此需要将这些服务器漏掉的访问请求加入到用户会话中，为 Web 日志挖掘提供完整的数据源。

6.结论

本文设计了一个基于大数据平台的可视化 Web 日志挖掘系统。在 Django 的 MVC 框架基础上，借助开源的 Bootstrap 框架，实现了一个面向用户的 Web 日志存储和挖掘系统。本章详细介绍了各个模块功能的内部实现细节，并展示了语言系统的整体框架和各个模块的具体信息。利用该系统，用户可以通过简单的前端操作实现本文提出的 Web 日志存储、聚类分析和关联规则挖掘。

鸣谢

感谢以下基金对本文的支持。兰州职业技术学院科研项目"'互联网+'视角下基于产教融合的高职院校创新创业团队建设研究"。