DATA EXTRACTION BY INFORMATION PROCESSING FROM VARIOUS USER RECOMMENDED SYSTEMS

S. Kalaimani[1], Dr. R.Mala[2]

[1]Research Scholar, PG and Research Department of Computer Science, Marudupandiyar College, Thanjavur, Tamilnadu, India
[2]Asst.Prof & Research Advisor, Department of Computer Science

Abstract

There is several user recommendation systems are recently available but they are satisfying the user recommendation with the disturbance of unwanted data. So there is need for filtering the required data for user conformability. In this recommendation system, since they are growing up for various user needs. The user required information with unwanted data. The raw data transfer for information processing to classifying into separated groups, filtering the unwanted data, finally extracted the actual data as user recommended to make better decision to a particular standard. In earlier, the scientific community is concerned to increase the accuracy of different classification methods, and major achievements have been made so far. Cloud computing has become a feasible mainstream solution for data processing, storage and distribution. It confirms on demand, scalable, compute and storage capacity. So rich amount of data in cloud database or any other cloud file systems, for that Naïve Bayes and support vector machine (SVM) classification algorithms are used to discover knowledge.

Recommender Systems apply for the machine learning and data mining techniques for filtering unseen information and can predict whether a user needed the given resource or not. Collaborative filtering recommender systems recommend items by identifying other users with same requirements and use their opinions for recommendation; such as content-based recommender systems recommend items based on the user needed information of the items. These kinds of various systems suffer from scalability, data parity, over specialization, and cold-start problems resulting in poor quality recommendations and lowest coverage. Hybrid recommended systems combine the individual systems to avoid certain limitations of these systems.

Keywords: Information Processing, Recommender System, Big Data, MongoDB.

1. Introduction

The large size of data generated in all fields of science is increasing extremely fast. MapReduce frameworks such as Hadoop are becoming a common and reliable choice to tackle the so called big data challenge. There are two main steps in the supervised classification process. The first process is the training step where the classification model is built. The second is the classification itself and it applies the trained model to assign unknown data to one out of a given set of class labels. The training step is the one that draws more scientific attention; it usually relies on a small identifiable data set that does not represent an issue for big data applications. Thus, the big data challenge affects mostly the classification step.

Hadoop MapReduce is a parallel programming technique for distributed processing environment, and it implemented on top of HDFS. The Hadoop MapReduce engine consists of a JobTracker and several TaskTrackers. When a MapReduce job is executed, the JobTracker splits it into small size of tasks (map and reduce) handled by the

TaskTrackers. In the Map step, the master node takes the given input, divides it into smaller sub-tasks and distributes them to worker nodes. Each worker node processes a sub-task and writes its results as key/value pairs. In the Reduce processing step, the values with the same key are grouped and it processed by the similar machine to form the final output.

The overlap of the fields of Natural Language Processing (NLP), where the goal is enabling computers to derive meaning or information from natural language, and Machine Learning, where systems that can learn from data are constructed, is diverse. The applications of combining the two fields ranges from marketing analysis and neutrality - assurance to more futuristic ideas like dynamic, affectively-aware systems. Such an overlap is the field of sentiment analysis, where NLP and machine learning are used in determining subjective information such as the tone of text. An important task in sentiment analysis is classifying text based on this information. To this end, many methods are used. Keyword spotting categories text by affect that category based on the presence and count of affect words. Of statistical methods, Support Vector Machines is "bunch of words" are highly often used. Support Vector Machines work by representing the inputs as points in space with gap that is as wide as possible between them. It predicts the class of new inputs based on which side of the gap they are mapped. A big problem is the scalability of algorithms to big data the ability of the system to accommodate large datasets. Our solution uses a bag of words representation, a naïve Bayes classifier, and Apache Spark, a distributed computing platform, to scale well.

2. Literature Review

2.1 Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier

This paper present a typical method to obtain valuable information is to extract the opinion from a message. Machine learning technologies are mostly used in sentiment classification because of their ability to "learn" from the training dataset to predict the decision making with high accuracy respectively. When the dataset is large, some algorithms might not scale up well. In this paper, we aim to evaluate the scalability of Naive Bayes classifier (NBC) in large datasets. Instead of using a standard library i.e., Mahout, we implemented NBC to reach fine-grain control of the analysis procedure. A Big Data analyzing system is also design for the process of study. The result is motivated to develop in that the accuracy of NBC is improved and approaches 82% when the dataset size increases. We have demonstrated that NBC is able to scale up to analyze the sentiment of millions movie reviews with increasing throughput.

2.2 Performance Evaluation of a MongoDB and Hadoop Platform for Scientific Data Analysis

Scientific services such as the Advanced Light Source (ALS) and Joint Genome Institute and projects such as the Materials Project have a growing up to capture, store, and analyze dynamic semi-structured data and metadata. A similar growth of semi-structured data within large Internet service providers led to the creation of NoSQL data stores for scalable indexing and MapReduce for scalable parallel analysis. MapReduce and NoSQL stores have been applied to scientific type of data. Hadoop is the most popular open source implementation of MapReduce, used to evaluated, utilized and modified for the addressing the requirements of different scientific analysis problems. ALS and the Materials Project are using MongoDB, a document oriented NoSQL store. However, there is a limited understanding of the performance trade-offs of using these two technologies together. In this paper we evaluate the performance, scalability and fault-tolerance of using MongoDB with Hadoop, towards the goal of identifying the right software environment for scientific data analysis.

2.3 MapReduce: Simplified Data Processing on Large Clusters

MapReduce is a programming model and an associated implementation for processing and generating large data sets. Users specify a *map* function that processes a key/value pair to generate a set of intermediate key/value pairs, and a *reduce* function that merges all intermediate values associated with the same intermediate key. Many real world tasks are expressible in this model, as shown in the paper.
Programs written in this functional style are automatically parallelized and executed on a large cluster of commodity machines. The run-time system takes care of the details of partitioning the input data, scheduling the program's

execution across a set of machines, handling machine failures, and managing the required inter-machine communication. This allows programmers without any experience with parallel and distributed systems to easily utilize the resources of a large distributed system.

2.4 Implementation of Map-Reduce based Context-Aware Recommendation Engine for Social Music Events

In this paper present modern ubiquitously connected world the amount of ever available product and service information within our daily lives is exploding. Powerful client devices, such as smart phones and tablets allow the users to get access to an unlimited amount of information on every product or service available. As the amount of available information on products by far exceeds the user's time to examine and filter detailed pieces of information in every situation, we expect that client-centric and context-aware information filtering is one of the thriving topics within the next years. A popular approach is to combine context-awareness with traditional recommendation engines in order to evaluate the relevance of a large amount of items for a given user situation. The goal is to proactively evaluate the situation of a user in order to automatically propose relevant products. Within this work we describe a general approach and the implementation of a software framework that combines traditional recommendation methods with a variable number of context dimensions, such as location or social context. The main contribution of this work is to show how to use a MapReduce programming model for aggregating the necessary information for calculating fast context-aware recommendations as well as how to overcome a typical cold start problem. The use-case at the end of this work evaluates the practical benefit of our general framework to introduce a client-centric, MapReduce based recommendation engine for real-time recommending music events and festivals.

2.5 Automatic Sentiment Analysis for Unstructured Data

Now-a-days Big Data have been created lot of buzz in technology world. Sentiment Analysis or opinion mining is very important application of 'Big Data'. Sentiment analysis is used for knowing voice or response of crowd for products, services, organizations, individuals, movie reviews, issues, events, news etc... In this paper we are going to discuss about exiting methods, approaches to do sentimental analysis for unstructured data which reside on web. Currently, Sentiment Analysis concentrates for subjective statements or on subjectivity and overlook objective statements which carry sentiment(s). So, we propose new approach classify and handle subjective as well as objective statements for sentimental analysis.

2.6 Crab: A Recommendation Engine Framework for Python

Crab is a flexible, fast recommender engine for Python that integrates classic information filtering recommendation algorithms in the world of scientific Python packages (NumPy, SciPy, and Matplotlib). The engine aims to provide a rich set of components from which you can construct a customized recommender system from a set of algorithms. It is designed for scalability, flexibility and performance making use of scientific optimized python packages in order to provide simple and efficient solutions that are accessible to everybody and reusable in various contexts: science and engineering. The engine takes users' preferences for items and returns estimated preferences for other items. For instance, a web site that sells movies could easily use Crab to figure out, from past purchase data, which movies a customer might be interested in watching to. This work presents our inactive in developing this framework in Python following the standards of the well-known machine learning toolkit Scikit-Learn to be an alternative solution for Mahout Taste collaborative framework for Java. Finally, we discuss its main features, real scenarios where this framework is already applied and future extensions.

2.7 SRSH: A Social Recommender System based on Hadoop

Online Social Networks (OSNs) accumulate a large amount of user-generated data and Social Recommender Systems (SRSs) can help users discover information they are interested. However, most of the existing SRSs do not have good scalabilities to process huge volumes of data. Aiming to this problem we design a social recommender system named SRSH, which is based on Hadoop parallel computing platform. SRSH provides second-degree friends, similar users, user community and content recommendation modules, which can meet user needs of finding potential friends and attractive content. Especially, every core methods existing in these modules above can be implemented using MapReduce parallel programming framework and run in Hadoop cluster. We have conducted extensive related experiments on the realistic dataset and the experimental results show that SRSH scales well and has the ability of dealing with the problem of recommendation in the large-scale OSN.

2.8 Performance Analysis of Book Recommendation System on Hadoop Platform

They consider the recommendation engines are computational intensive and hence ideal for Hadoop Platform. This research paper aims at building a book recommendation engine which uses data mining for recommending books. It will give its users' the ability to upload and download engineering books as well as novels which will be used to draw out conclusions about the stream of a user and the genre of the books liked by that user. It will analyze the user behavior by making use of content based filtering and will apply association rules in data mining for displaying individualized recommendation system of books. This database will then be transferred to the Hadoop HDFS so that a comparison between the read and write efficiency between the local system and Hadoop can be made.

2.9 An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering

They consider the recommender systems apply machine learning and data mining techniques for filtering unseen information and can predict whether a user would like a given resource. To date a number of recommendation algorithms have been proposed, where collaborative filtering and content-based filtering are the two most famous and adopted recommendation techniques. Collaborative filtering recommender systems recommend items by identifying other users with similar taste and use their opinions for recommendation; whereas content-based recommender systems recommend items based on the content information of the items. These systems suffer from scalability, data parity, over specialization, and cold-start problems resulting in poor quality recommendations and reduced coverage. Hybrid recommender systems combine individual systems to avoid certain aforementioned limitations of these systems. In this paper, we proposed a unique switching hybrid recommendation approach by combining a Naive Bayes classification approach with the collaborative filtering. Experimental results on two different data sets show that the proposed algorithm is scalable and provide better performance–in terms of accuracy and coverage–than other algorithms while at the same time eliminates some recorded problems with the recommender systems.

2.10 Thumbs up? Sentiment Classification using Machine Learning Techniques

They consider the problem of classifying documents not by topic, but by overall sentiment, e.g., determining whether a review is positive or negative. Using movie reviews as data, we find that standard machine learning techniques definitively outperform human-produced baselines. However, the three machine learning methods we employed (Naive Bayes, maximum entropy classification, and support vector machines) do not perform as well on sentiment classification as on traditional topic-based categorization. We conclude by examining factors that make the sentiment classification problem more challenging.

3. Discussion

This section reports some experiments conducted upon various tools. These experiments were carried out on the Mortar platform, a cloud-computing, open-source framework for organizing, developing, testing, and deploying big data processing applications based on Hadoop. This platform relies on MapReduce, which uses the Hadoop framework to distribute the data and processing across a resizable Compute Cloud cluster, and on Amazon Simple Storage Service. On Mortar one can work directly with Pig on Hadoop and configure the number of cluster nodes on which the Pig Latin script will be executed in a simple and flexible way.

4. Conclusion

In this paper, different tools, framework and databases are presented, a tool able to perform classification processes on huge amounts of data, exploiting the benefits of working on clusters with the Hadoop framework. An experimental analysis indicated that the speedup achieved by the tool increases with the amount of data being processed. Additionally, the results showed that increasing the number of nodes in the cluster does not necessarily provide a corresponding reduction of execution times. Thus, the proper cluster configuration depends not only on the operations to be executed but also on the amount of input data; there must be a balance between the amount of data to be processed and the number of nodes to be used to achieve the best performance. Finally, it provides result of various user recommender systems with Big Data environment.

REFERENCES

[1] S. Vijaykumar, M. Balamurugan, S.G. Saravanakumar, Unique Sense: Smart Computing Prototype, Procedia Computer Science, Volume 50, 2015, Pages 223-228, ISSN 1877-0509, http://dx.doi.org/10.1016/j.procs.2015.04.056.

[2] V. A. Ayma, "Classification Algorithms for Big Data Analysis , A Map Reduce Approach", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XL-3/W2, 2015 PIA15+HRIGI15 – Joint ISPRS conference 2015, 25–27 March 2015.

[3] Vijaykumar S, Dr. M. Balamurugan, Ranjani K, Big Data: Hadoop Cluster Deployment on ARM Architecture, International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE), Vol. 4, Special Issue 1, June 2015, ISSN 2278-1021 & 2319-5940.

[4] Bingwei Liu, "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier", 2013 IEEE Conference on Big Data, Oct 2013, pp. 99-104, INSPEC Accession Number: 13999322, 10.1109/BigData.2013.6691740.

[5] Vijaykumar, S., Saravanakumar, S., & Balamurugan, M. (2015). Unique Sense: Smart Computing Prototype for Industry 4.0 Revolution with IOT and Bigdata Implementation Model. *Indian Journal Of Science And Technology, 8*(35). doi:10.17485/ijst/2015/v8i35/86698.

[6] Elif Dede, "Performance evaluation of a MongoDB and Hadoop platform for scientific data analysis", Proceeding Science Cloud'13, Proceeding of 5 the 4th ACM workshop on Scientific Cloud Computing, ages 13-20, 10.1145/2465848.2465849.

[7] S.Vijaykumar, S. G. Saravanakumar . Future Robotic Memory management, Advances in Digital Image Processing and Information Technology Communications in Computer and Information Science .Volume 205, 2011, pp 315-325. ISSN : 1865-0929. DOI:10.1007/978-3-642-24055-3_32.

[8] Mustansar Ali Ghazanfar and Adam Pr gel-Bennett, "An Improved Switching Hybrid Recommender System Using Naive Bayes Classifier and Collaborative Filtering", *in Proceedings of the International MultiConference of Engineers and Computer Scientists* Vol-I, Hong Kong, IMECS March 17–19, 2010.

[9] Vijay Kumar, S., Saravanakumar, S.G.: Revealing of NOSQL Secrets. CiiT Journal 2(10), 310–314 (2010), Url= http://www.ciitresearch.org/dmkeoctober2010.html.

[10] P. Chesley, B. Vincent, L. Xu, and R. K. Srihari, "Using verbs and adjectives to automatically classify blog sentiment ", *Training, vol. 580, no. 263*, pp. 233

[11] M. Thomas, B. Pang, and L. Lee, "Get out the vote: Determining support or opposition from congressional floor-debate transcripts", *in Proceedings of the 2006 conference on empirical methods in natural language processing.* Association for Computational Linguistics, 2006, pp. 327–335.

[12] P. Domingos and M. Pazzani, "On the optimality of the simple bayesian classifier under zero-one loss", *Machine learning, vol. 29* no. 2-3, pp. 103–130, 1997.

[13] S. Vijaykumar and S. Saravanakumar, "Future Robotics Database Management System along with Cloud TPS," *Intl. Journal on Cloud Computing: Services&Architecture (IJCCSA)*, pp. 431–438, 2011.

[14] H. Karloff, S. Suri, and S. Vassilvitskii, "A model of computation for Map Reduce", *in Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms.* Society for Industrial and Applied Mathematics, 2010, pp. 938–948.