International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 5 Issue: 7 613 – 616

Classifying User Predilections using Naïve Bayes Classifier (NBC) and Jaccard Similarity for Service Recommender System in Big Data Applications

Dr.R.Mala
Assistant Professor and Research Advisor
Department of Computer Science, Marudupandiyar College
Thanjavur, Tamilnadu, India

S. Kalaimani
Research Scholar, Department of Computer Science
Marudupandiyar College
Thanjavur, Tamilnadu, India

Abstract-Service recommender systems have been shown as valuable tools for providing appropriate recommendations to users. The main objective is to identify a system that will classify the user reviews using effective methods and provide personalized recommendations to the users. The proposed architecture will present the different ratings and rankings of services to different users by considering diverse users' preferences, and therefore it will meet users' personalized requirements. The data classification can be achieved through analysing the user review as positive or negative using Naive Bayes classifier (NBC) in large-scale datasets and Jaccard Similarity and MinHash used to compute the similarity and provide the recommendation to user.

Keywords: *Big data, NoSQL, Polarity Classification, Parallel Processing, Recommender System.*

*****

I. INTRODUCTION

In the recent trends, online Ecommerce is being getting popular among all the internet users and they are being attracted towards the new offers and discounts. This leads to the Ecommerce providers to improve their systems to provide right product listing based on the user interests. Recommendation systems plays vital role in the Ecommerce market by suggesting the uses to buy relevant products.[2] Until recently, users commonly asked for a recommendation from their own circle of known friends or family. However, recommendations demand a certain level of trustworthy knowledge and not everyone is eligible to provide a skilled recommendation. These systems build with data mining and machine-learning algorithms that will be changing day by day based on the upcoming products and latest fashion trends. The ability to automatically mining useful information from massive data has been a common concern for organizations who own large datasets. This leads to the need for identifying scalable framework, which can works well on the dynamic environment. The Map Reduce framework is commonly used to analyze large datasets that works well in distributing the data's into multiple distinguishable splits and merging the results into commodity servers. [5][14]The framework provides a simple and powerful interface for programmers to solve large-scale problems using a cluster of commodity computers [6][7][18].

A common method to extract the valuable information is to identify the two corners (Both positive and negative) of the feedback from the message. This will be majorly used by the manufacturers to decide whether they can proceed with their product or not. Naive Bayes used to for textual categorization and best to implement in the highly scalable environment [9] [17].

In our previous paper, categorized the user preference as positive and negative using Naive Bayes classifier (NBC) in large-scale datasets. Shown that the suitable Recommender System to improve the results when the dataset size increases [18] [10] [14] [12].

In this paper we will be covering Jaccard similarity and MinHash Values measures and how it will improve the overall system performance. The details of implementing Jaccard similarity.

II. BACKGROUND

Recommender system provides appropriate recommendation to users when user uses online service on web. Traditional method is inefficient when processing or analyzing large volume of data. Existing recommender system considers user preference and it is recommended to the user using relational database and linear processing. In this system, the user preference is categorized as positive and negative for recommendations to user using NoSQL (Not Only SQL) database and parallel processing for recommender system. Machine learning technologies are used to classify the user preference because of their ability to learn from the training dataset to predict or support decision making with relatively high accuracy. The user preferences are various types such as unstructured, semi-structured and it is complex to manage the relational databases so that NoSQL database is used to process this type of dataset. NoSQL database provide Non-relational and schema-less data model, Low latency and high performance, highly scalable used to manage the user preference for classification purpose as well as recommendation to user. As the preferences are categorized

it improves the accuracy of service recommender system using NoSQL databases and Parallel Processing [3].

MongoDB is an open source NoSQL document store database, commercially supported by 10gen. Although MongoDB is non-relational, it implements many features of relational databases, such as flexible data models, auto-Sharding and load balancing, sorting, secondary indexing and range queries. MongoDB does not organize data in tables with columns and rows. Instead, data is stored in documents, each of which is an associative array of scalar values, lists, or nested associative arrays. MongoDB documents are serialized naturally as JavaScript Object Notation (JSON) objects, and are in fact stored internally using a binary encoding of JSON called BSON. To scale its performance on a cluster of servers, MongoDB uses a technique called sharing, which is the process of splitting the data evenly across the cluster to parallelize access [13].

III. PROPOSED METHOD

A. *DataFlow*

In the data flow is to describe the how data's are flow from starting to at the end the process during this process there are three main process are there that is polarity classification, similarity computation and calculate rating such a things. Finally the recommendations are generated based on the rating [12]

In our previous paper shown that through Naive Bayes classification on top of Hadoop framework a significant result have been achieved to add further more in the existing setup added Jaccard similarity with MinHash value measure in identifying the similarity computation. [18][1].

B. *Jaccard Similarity and MinHash Values Measure*

The Jaccard similarity coefficient is a commonly used indicator of the similarity between two sets. For sets A and B it is defined to be the ratio of the number of elements of their intersection and the number of elements of their union:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

This value is 0 when the two sets are disjoint, 1 when they are equal, and strictly between 0 and 1 otherwise. Two sets are more similar (i.e. have relatively more members in common) when their Jaccard index is closer to 1. It is our goal to estimate $J(A,B)$ quickly, without explicitly computing the intersection and union.

Let h be a hash function that maps the members of A and B to distinct integers, and for any set S define $h_{min}(S)$ to be the minimal member of S with respect to h that is, the member x of S with the minimum value of $h(x)$. Now, if we apply h_{min} to both A and B, we will get the same value exactly when the element of the union $A \cup B$ with minimum hash value lies in the intersection $A \cap B$. The probability of this being true is the ratio above, and therefore:

$$Pr[h_{min}(A) = h_{min}(B)] = J(A,B)$$

That is, the probability that $h_{min}(A) = h_{min}(B)$ is true is equal to the similarity $J(A,B)$, assuming randomly chosen sets A and B. In other words, if r is the random variable that is one when $h_{min}(A) = h_{min}(B)$ and zero otherwise, then r is an unbiased estimator of $J(A,B)$, r has too high a variance to be a useful estimator for the Jaccard similarity on its own it is always zero or one. The idea of the MinHash scheme is to reduce this variance by averaging together several variables constructed in the same way [18].

After the Preprocessing the dataset are applied for following sequence for parallel processing using MongoDB Map Reduce framework shown in Fig. 2. Similarity Computation this is perform two process Finding Users Similar to a Particular User and Finding Number of Movies that Belong to each Genre. Calculate Rating such as Finding Average Rating of Movies. Generate Recommendation such as Recommendation System Based on Jaccard Similarity.

C. *JS M using MHF on Map Reduce*

Defining the Problem: The map reduce version deals with each user individually instead of dealing with each movie. For that we produce hash values for each *user, movie* at the map stage and then using reduce function we find the minimum value for each hash function and set the outcome of the hash function for that user to the minimum value produced by map function.

Now we find hash values of a particular user that we are interested to find similar people to. Find similarity measure of other users to the particular user we are interested.

Algorithms: The method is used on finding minhash signatures for each user.

The probability that the minhash function for a random permutation of rows produces the same value for two sets equals the Jaccard similarity of those sets.

For each movie (m) we have:

1. Compute $h_1(m)$, $h_2(m)$, ... , $h_n(m)$.
2. For each user u do the following:
   a) If u has rated movie m below 2.5 do nothing.
   b) However, if u has rated movie m above 2.5, then for each i = 1,2,...,n set $SIG(i,u)$ to the smaller of the current value of $SIG(i,u)$ and $h_i(m)$.

Choose first user u^0 that has approximate Jaccard similarity measure equal to 1. This is an approximation and convergence depends on number of hash functions and the number of functions must be higher for reasonable results. Now we print movies of both users and find those movies that are rated high by u^0 and suggest to u.

IV. RESULT AND DISCUSSION

The result statistics include the computation time and the throughput of the system. Table I shows that the recommendation processing time for every movie dataset in our Hadoop NBC, MongoDB Jaccard Similarity program as decreases when the dataset size increases. A dataset of 2K reviews did not benefit from the parallelization of Hadoop because the input data is smaller than the size of one block in HDFS. Although three replicas are distributed in different nodes, there are at most three nodes in the Hadoop cluster that can access the input data locally. After the input data increasing to a certain size, the advantage of Hadoop starts to appear in that the processing time for the same amount of reviews is drastically reduced compare to the 2K case.

TABLE I: Recommendation Processing Time with respect to movie dataset size

Second/10k Recommendations	*Dataset Size(k)*
350.1	5
38.20	30
5.34	100
5.13	200
4.55	300
4.23	400
3.40	500

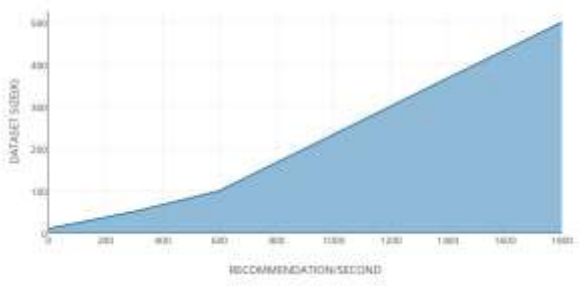

Fig. 2: Recommendation Throughput of the System with respect to dataset size

To observe the results in another dimension, Fig. 6 shows the throughput of the system with respect to the size of dataset. The number of movie reviews that the system can processes in one second increases from 600 (100K case) to 1800 (500K case).

Overall, our implementation of NBC and JSM using MHF is able to scale up to two million reviews sampled from the Amazon dataset and Movie Lens. The Throughput tends to stable when the dataset size increases. These results are based on the simple processing of review texts, Movie Lens dataset. Further filtering of the input data might be able to increase the throughput accuracy.

V. REFERENCES

[1] M. Gamon, A. Aue, S Corston-Oliver, and E. Ringger, "Pulse: Mining customer opinions from free text", in Advances in Intelligent Data, Analysis VI, ringer, 2005, pp.221-132.

[2] Bingwei Liu, Erik Blasch, Yu Chen, Dan Shen and Genshe Chen, "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier", Big Data, 2013 IEEE International Conference on IEEE, Oct, 2013, pp. 99–104.

[3] 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.

[4] Improved Consistent Sampling, Weighted Minhash and L1 Sketching Published in: Data Mining (ICDM), 2010 IEEE 10th International Conference Print ISBN: 978-1-4244-9131-5 Electronic ISBN: 978-0-7695-4256-0 Print on Demand(PoD) ISBN: 978-1-4244-9131-5.

[5] Adaptive near-duplicate detection via similarity learning Authors: Hannaneh Hajishirzi University of Illinois at Urbana-Champaign, Urbana, IL, Proceeding SIGIR '10 Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval Pages 419-426 Geneva, Switzerland — July 19 - 23, 2010 ACM New York, NY, USA ©2010 table of contents ISBN: 978-1-4503-0153-4.

[6] S. Vijaykumar, SG Saravanakumar, "Implementation of NOSQL for robotics", Emerging Trends in Robotics and Communication Technologies (INTERACT), 2010, Doi: 10.1109/INTERACT.2010.5706225.

[7] Instance-Based Matching of Large Ontologies Using Locality-Sensitive Hashing Songyun DuanAchille FokoueOktie HassanzadehAnastasios KementsietsidisKavitha SrinivasMichael J. Ward DOI: 10.1007/978-3-642-35176-1_4

[8] Charikar, M.: Similarity estimation techniques from rounding algorithms. In: ACM Symp. on Theory of Computing (STOC), pp. 380–388 (2002)

[9] 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

[10] 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.

[11] A. Kennedy and D. Inkpen, "Sentiment classification of movie reviews using contextual valence shifters", Computational Intelligence, vol. 22, no. 2, 2006 pp. 110–125.

[12] J. Dean and S. Ghemawat, "Map Reduce: simplified data processing on large clusters ", Communications of the ACM vol. 51, no. 1 pp. 107– 113, 2008.

[13] D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval", in Proceedings of the 20th international conference on World Wide Web. Machine Learning: ECML-98 pp. 4–15, 1998.

[14] 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.

[15] Dr.M. Balamurugan, Saravanakumar S.G, and Vijaykumar S,Unique Sence: Smart Computing Prototype for Industry 4.0 Revolution with IOT and Big Data Implementation Model, Indian Journal of Science and Technology, (35), 1-8, 2015.

[16] A. Nancy, Dr.M. Balamurugan, S. Vijaykumar, "A Comparative Analysis of Cognitive Architecture", International Journal of Advanced Research Trends in Engineering and Technology (IJARTET), Vol.3, Special Issue 20, April 2016, Pg. 152-155, ISSN (P): 2394-3777, ISSN (O): 2394-3785.

[17] M. Mayilvaganan, D. Kalpanadevi. Comparison of Classification Techniques for predicting the Cognitive Skill of Students in Education Environment. 2014 IEEE International Conference on Computational Intelligence and Computing ResearchIEEE Xplore. Advancing Technology for Humanity. IEEE Catalog Number: CFP1420J-PRT. ISBN: 978-1- 4799-3974-9. Page: 279-282. (Thomson Reuters and Scopus Indexed)

[18] Ghosh K, Indra N. Efficacy of Centella asiatica ethanolic extract on cadmium induced changes in blood haematology, hepatic and nephritic function markers in albino rats. Int J Curr Res 2015 07(07): 18283-18288. [ISSN 0975-833X]

[19] Characterization of User Inclinations for Service Recommender System in Big Data Applications S. Kalaimani1, Dr. R. Mala2, International Journal of Innovative Research in Science, Engineering and Technology (An ISO 3297: 2007 Certified Organization) Vol. 5, Issue 11, November 2016 ISSN(Online) : 2319-8753 ISSN (Print) : 2347-6710

[20] Doan, A., Domingos, P., Halevy, A.Y.: Reconciling Schemas of Disparate Data Sources: A Machine-Learning Approach. In: ACM SIGMOD Int'l Conf. on Mgmt. of Data, pp. 509–520 (2001)

[21] Doan, A., Halevy, A.Y.: Semantic Integration Research in the Database Community: A Brief Survey. AI Magazine 26(1), 83–94 (2005) Google Scholar.

[22] S.G. Saravanakumar, S. Vijaykumar, "Eevealing of NOSQL Secrets" CiiT International of Data Mining Konwledge Engineering 2, 310-314, 2010.