Science, Engineering & Education, 2, (1), 2017, 58-65

A framework for cloud based hybrid recommender system for big data mining

Kamal Al-Barznji, Atanas Atanassov*

University of Chemical Technology and Metallurgy, 8 Kl. Ohridski, 1756 Sofia, Bulgaria

Received 27 September 2017, Accepted 27 October 2017

ABSTRACT

The modern web platforms dealing with large number of items are using recommender systems to suggest automatically new interesting items to users and, hence, to keep them using the platform. From the users' perspective, recommender systems help them to handle information. In this paper, a Framework for Cloud Based Hybrid Recommender System (FCHRS) for Big Data Mining is proposed and methods and algorithms that are used in the framework are discussed. It is based on the Iterative collaborative filtering, which traditionally is the most used approach, and on the Sentiment Analysis (known as opinion mining) as well. It refers to the use of natural language processing, text analysis and computational linguistics to identify and to extract subjective information from source materials. Thus it becomes essential for the enterprise to mine social media data (big data) to make recommendations. The combination of results of two algorithms provides more useful business intelligence. This combination is new and unexplored area of research.

Keywords: Big Data Mining, Cloud Computing, Big Data Platform, Recommendation System, Iterative Collaborative Filtering, Sentiment Analysis.

INTRODUCTION

Enterprises in the real world can mine such data to gain comprehensive and actionable knowledge. This process is known as big data analytics or big data mining [1]. Big Data Mining refers to the activity of going through big datasets to look for relevant information. Big Data mining is the capability of extracting useful information from these large datasets or streams of data. The extracted knowledge is very useful and the mined knowledge is the representation of different types of patterns and each pattern corresponds to knowledge [11]. Big Data analysis is one of the most important issues in data mining research nowadays. Big Data becomes greatly important when high calculations have to be processed or the amount of data is greater than the capacity of the system. There are three features of big data: volume, velocity and variety. Volume indicates

**Correspondence to: Atanas Atanassov, University of Chemical Technology and Metallurgy, 8 Kl. Ochridski, 1756 Sofia, Bulgaria, E-mail: naso@uctm.edu*

the amount of data that needs to be processed. Could it be moved to zeta bytes? Velocity refers to the speed at which data can be processed with minimal error rate. Variety aims to all types of data for example unstructured, semi structured and structured data. Facebook and Twitter logs, text files, word documents, etc., are raw form data called unstructured data.

Large user feedback data is being produced every day in various areas such as movies, food, electronic, etc. The amount of user data is increasing dramatically due to the growth of e-commerce websites. At this point it raises the importance of topics like how can be stored and how to understand all these data. Big data is one of the best solutions of storing these data and has motivated the interest of recommendation systems to understand the data. Reviews have become a very effective way for people to share expectations of a product or services, to express their opinions, to support their products or even to educate each other's by publishing textual data. The meaningful and useful information obtained from user reviews by sentiment analysis can be used to improve recommender systems. Researchers have argued that recommendation systems could be very beneficial. By explaining why a product is recommended, it helps users to make better and faster decisions, and help users to buy or enjoy. The most common way of recommendation is collaborative filtering (CF) methods to generate user specific recommendation of an item based on patterns of ratings or usage without need for outside information about both item and user [2]. The growth of e-commerce sites and online businesses are enhancing the requirements of a robust recommendation system [3]. Recommender systems are now pervasive in consumers' lives. They aim to help users in finding items that they would like to buy or consider based on huge amounts of data collected. Amazon, Facebook, LinkedIn, and other commercial and social networking websites use these systems [4].

CLOUD COMPUTING

In the present day software era the ever growing data demands elastically scalable data centers which can be conveniently accessible with high quality in a secure way. This demand led to cloud computing as one of the emerging technologies of today. Cloud computing releases computer services, computer software, and data storage away from locally hosted infrastructure and into cloud-based solutions. The ability of the cloud services to provide apparently an unlimited supply of computing power on demand to users has caught the attention of industry as well as academia. In the past couple of years software providers have been moving more and more applications to the cloud. Several big companies such as Amazon, Google, Microsoft, Yahoo, and IBM are using the cloud. Today, forward-thinking business leaders are using the cloud within their enterprise data centers to take advantage of the best practices that cloud computing has established, namely scalability, agility, automation, and resource sharing. By using a cloud-enabled application platform, companies can choose a hybrid approach to cloud computing that an organization's existing infrastructure to launch new cloud-enabled applications. This hybrid approach allows IT departments to focus on innovation for the business, reducing both capital and operational costs and automating the management of complex technologies. Cloud services claim to provide nearly everything needed to run a project without owning any IT infrastructure. From e-mail, Web hosting to fully managed applications resources can be provided on–demand. This helps to reduce development cost and hardware cost [5].

RECOMMENDATION SYSTEM

Recommendation is a suggestion that can help in making good decisions faster. Recommendation system provides the facility to understand a person's taste and find new, desirable content for

them automatically. Although people's tastes vary, they do follow patterns. People tend to like things that are similar to other things they like as well as other similar behaviour person likes. In fact these are the two broadcast categories of recommender engine algorithms: user-based and item-based recommenders. These recommendations are based on two filtering techniques namely collaborative and content-based filtering techniques. Collaborative filtering produces recommendations based on, and only based on, knowledge of users relationship with items, product and services. These techniques require no knowledge of properties of items and characteristics. Content-based recommendations are based on attributes of items. Here suggestions are based on the content related to items and their aspects. One more approach has become more popular where both the filtering techniques can be applied on different levels of recommendation system, called hybrid filtering technique [3].

Hybrid Recommender System

It is natural to consider combining several different recommender algorithms into a hybrid recommender system. In some applications, hybrids of various types have been found to outperform individual algorithms. Hybrids can be particularly beneficial when the algorithms involved cover different use cases or different aspects of the data set. For example, item-item collaborative filtering suffers when no one has rated an item yet, but content-based approaches do not. A hybrid recommender could use description text similarity to match the new item with existing items based on metadata, allowing it to be recommended anyway, and increase the influence of collaborative filtering as users rate the item; similarly, users can be defined by the content of the items they like as well as the items themselves [6].

Examples of Recommendation Engines

LinkedIn, the business-oriented social networking site, forms recommendations for people you might know, jobs you might like, groups you might want to follow, or companies you might be interested in. LinkedIn uses Apache Hadoop to build its specialized collaborative-filtering capabilities. Amazon, the popular e-commerce site, uses content-based recommendation. When you select an item to purchase, Amazon recommends other items other users purchased based on that original item. Amazon patented this behaviour, called item-to-item collaborative filtering. Hulu, a streaming-video website, uses a recommendation engine to identify content that might be of interest to users. It also uses (offline) item-based collaborative filtering with Hadoop to scale the processing of massive amounts of data. Details of Hulu's online and offline ItemCF architecture are publicly available. Netflix, the video rental and streaming service, is a famous example. In 2006, Netflix held a competition to improve its recommendation system, Cinematch. In 2009, three teams combined to build an ensemble of 107 recommendation algorithms that resulted in a single prediction. This ensemble proved to be the key to improving predictive accuracy, and the combined team won the prize. Other sites that incorporate recommendation engines include Facebook, Twitter, Google, MySpace, Last.fm, Del.icio.us, Pandora, Goodreads, and your favorite online news site. Use of a recommendation engine is becoming a standard element of a modern web presence [15].

PROPOSED SYSTEM

There are scores of users purchasing online. You are one of them. When you try to choose some category of products, the system can mine big data pertaining to other users (cross of other users who chose such product earlier) and give valuable recommendations, e.g. when you enter into a big shop to purchase clothes. You inform the sales person for your requirements. Immediately he/she recommended best one that you wanted. You are happy and purchase items without wasting time.

In this paper, a Framework for Cloud Based Hybrid Recommender System (FCHRS) for Big

Data Mining (the large amount data) is proposed. It is available on the web in the form of ratings, reviews, opinions, complain, remarks, feedback, and comments about any item (product, event, individual and services). Fig. 1 shows architecture for Big Data Mining for Generating Recommendations.

Our framework takes big data as input. Our recommender system employs one or more recommendations algorithms to generate recommendations. The recommendations are based on different kinds like: Based on place or price or range or offers or popularity or any other attribute. Two algorithms for generating recommendations are used: First, a novel iterative collaborative filtering algorithm on big data for generating recommendations (generally made from relational data); and second, a novel sentiment analysis algorithm is used to generate recommendations from big data.

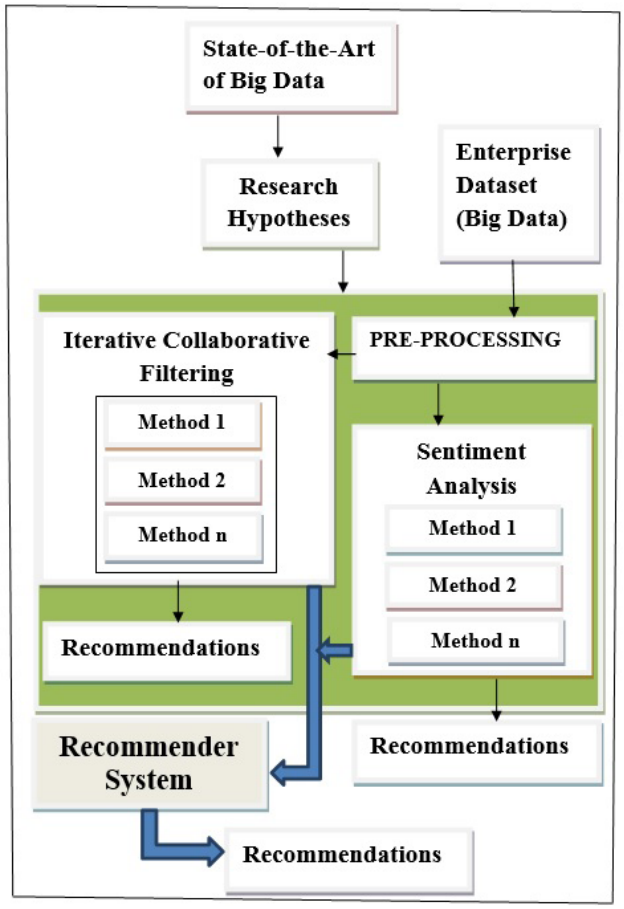

Fig. 1. Overview of Proposed Architecture for Big Data Mining for Generating Recommendations.

This is generally made from textual data. The first algorithm may work on the enterprise's historical data. These data are in the form of tables (structured). But the second algorithm might use social media data related to an enterprise (big data) which is unstructured.

An enterprise has data in data warehouse and can gain business intelligence and provide recommendations. But the company wants to get knowledge on customer behaviour on its products and services from social media where people have opinions. Thus it becomes essential for the enterprise to mine social media data (big data) to make recommendations. The combination of results of two algorithms provides more useful business intelligence. This combination is the new and unexplored area of research. It will be more useful for adding big value to enterprises.

Iterative Collaborative Filtering Algorithm

In this section the iterative collaborative filtering is presented. Collaborative filtering approaches build a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users; then thit model is used to predict items (or Ratings for items) that the user may be interested in [3].

Collaborative filtering methods analyze large amount of information about preferences of users and predict preferences of similar users for recommending items. In collaborative filtering method an accurate prediction of preferences of a user and recommendation of items is possible without any need for detailed analysis of item features. A basic assumption in collaborative filtering is that users would like similar kinds of items as they have liked in past. Collaborative filtering methods suffer from issues like - cold start, scalability and sparsity [9].

The pseudo-code is shown in **Algorithm 1.** Note that no need to do the rough estimate up-

date every time when r_{u0}, i_0 for a specific u_0 and i_0 are estimated. In fact, for multiple estimations in each iteration, it needs to go through this step only once. Therefore this process significantly improves the efficiency of the algorithm [7].

Algorithm 1 Iterative Collaborative Filtering

Given m users $u_{1,\ldots\ldots,}u_m$ and n items $i_{1,\ldots,}i_n$;

Given sparse rating matrix R_{mxn} in which $r_{u,i}$ denotes the rating of item i by user u.

Goal: Estimate $\boldsymbol{r_{u0,i0}}$ for specific $\boldsymbol{u_0}$ and $\boldsymbol{i_0}$

while $\left|\hat{r}_{u0,i0}^{new} - \hat{r}_{u0,i0}^{old}\right| \geq 0.01$ do

Calculate $\bar{r}_u = \frac{\sum_{i \in I_u} r_{u,i}}{|I_u|} \forall u \in U$

Calculate

$$s_{u,v} = \frac{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_{uv}} (r_{u,i} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_{uv}} (r_{v,i} - \bar{r}_v)^2}}$$

$\forall (u,v) \in UxU \; such \; that \; |I_{uv}| > M, zero \; otherwise.$

for $u = u_{1,\ldots\ldots,} u_m$ **do**

$\boldsymbol{d_u}$ =Number of available ratings on specific user $\boldsymbol{u}$

end for

for $i = i_{1,\ldots\ldots,} i_n$ **do**

$\boldsymbol{c_i}$ = Number of available ratings on specific item $\boldsymbol{i}$

end for

$$\gamma = \frac{3(m+n-1)}{m \; x \; n} \sum_{u \in U} d_u$$

for $u = u_{1,\ldots\ldots,} u_m$ **do**

for $i = i_{1,\ldots\ldots,} i_n$ **do**

if $d_u + c_i > \gamma$ and $r_{u,i}$ is missing **then**

Estimate $r_{u,i} = \bar{r}_u + \frac{\sum_{v \in U} s_{u,v} (r_{v,i} - \bar{r}_v)}{\sum_{v \in U} |s_{u,v}|}$;

end if

end for

end for

$$\hat{r}_{u0,i0} = \bar{r}_{u0} + \frac{\sum_{u \in U} s_{u0,u} (r_{u,i0} - \bar{r}_u)}{\sum_{u \in U} |s_{u0,u}|}$$

end while

return $\hat{r}_{u0,i0}$

Sentiment Analysis Algorithm

Sentiment Analysis, also known as opinion mining, is a powerful tool you can use to build smarter products. It is a natural language processing algorithm that gives you a general idea about the positive, neutral, and negative sentiment of texts. Social media monitoring applications and companies all rely on sentiment analysis and machine learning to assist them in gaining insights about mentions, brands, and products [12]._

Sentiment Analysis is refers to the use of natural language processing, text analysis and computational linguistics to identify and an extract subjective information in source materials. Also, this algorithm takes an input string and assigns a sentiment rating in the range [-1 to 1] (very negative to very positive) [13]. The internet is a resourceful place with respect to sentiment information. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites [8].

Example: Sentiment Analysis Using MapReduce Custom Counters: MapReduce jobs report results using a wide array of built-in counters. You can add your own counters and analyze the results in your applications. One use for custom counters is the sentiment analysis. To perform a basic sentiment analysis, you count up the positive words and negative words in a data set. Divide the difference by the sum to calculate an overall sentiment score [14].

Sentiment = (positive - negative) / (postitive + negative)

BIG DATA PLATFORM

Hadoop

Hadoop is an open-source Java framework for large-scale data processing and querying vast amount of data across clusters of computers. It is an Apache project initiated and led by Yahoo. It is mostly inspired by Google's MapReduce and GFS (Google File System). Hadoop is immensely

used by big companies like Yahoo, Facebook, Microsoft, and Amazon [1]. Hadoop is a large scale distributed batch processing infrastructure. It allows for the distributed calculation of large data sets called big data through clusters of computers using Map Reduce and HDFS file system. Hadoop can be used with data mining applications also recommendation algorithms by Mahout Apache project. The whole dataset is then transferred to the Hadoop File System and it uses a recommendation algorithm over frameworks [2].

Hadoop Distributed File System

Hadoop comes with a distributed file system called HDFS (Hadoop Distributed File system). HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. The Hadoop file system is designed for storing petabytes of a file with streaming data access using the idea that most efficient data processing pattern is a write-once, read many-times pattern. HDFS stores metadata on a dedicated server, called NameNode. Application data are stored on other servers called DataNodes. All the servers are fully connected and communicate with each other using TCP-based protocols [5].

Map Reduce

MapReduce is a programming model and software framework first developed by Google (Google's Map Reduce paper submitted in 2004). Map Reduce facilitates and simplifies the processing of big amounts of data in parallel on large scale clusters of commodity hardware in a robust, reliable and fault-tolerant manner. It can handle petabytes of data with thousands of nodes [1].

Mahout

Mahout is an open source project to provide free implementations of scalable and distributed machine learning algorithms in the areas of collaborative filtering, clustering and classification. It provides both non-distributed and distributed (Map-Reduce) algorithms for recommendation [9]. Map-reduce is a powerful technique for numerical computation and especially when you have to compute large data sets on Hadoop. The numerical computation is foundation of algorithms used to recommend. Cloudera Platform combines Hadoop framework and Mahout (algorithms) [15]. Mahout's recommenders expect interactions between users and items as input. The easiest way to supply such data to Mahout is in the form of a text file, where every line has the format userID, itemID, value. Here userID and itemID refer to a particular user and a particular item, and value denotes the strength of the interaction (e.g. the rating given to a movie) [16].

GOALS OF RECOMMENDER SYSTEMS

Before discussing the goals of recommender systems, the various ways in which the recommendation problem may be formulated are introduced. The two primary models are as follows:

• Prediction version of problem

The first approach is to predict the rating value for a user-item combination. It is assumed that training data is available, indicating user preferences for items. For m users and n items, this corresponds to an incomplete m × n matrix, where the specified (or observed) values are used for training. The missing (or unobserved) values are predicted using this training model. This problem is also referred to as the matrix completion problem because of incompletely specified matrix of values, and the remaining values are predicted by the learning algorithm.

• Ranking version of problem

In practice, it is not necessary to predict the ratings of users for specific items in order to make recommendations to users. Rather, a merchant may wish to recommend the top-k items for a

particular user, or determine the top-k users to target for a particular item. The determination of the top-k items is more common than the determination of top-k users, although the methods in the two cases are exactly analogous. Throughout this paper, only the determination of the top-k items are discussed, because it is the more common setting. This problem is also referred to as the top-k recommendation problem, and it is the ranking formulation of the recommendation problem. In the second case, the absolute values of the predicted ratings are not important. The first formulation is more general, because the solutions to the second case can be derived by solving the first formulation for various user-item combinations and then ranking the predictions. However, in many cases, it is easier and more natural to design methods for solving the ranking version of the problem directly.

In order to achieve the broader business-centric goal of increasing revenue, the common operational and technical goals of recommender systems are as follows:

Relevance: The most obvious operational goal of a recommender system is to recommend items that are relevant to the user at hand. Users are more likely to consume items they find interesting. Although relevance is the primary operational goal of a recommender system, it is not sufficient in isolation. Therefore, several secondary goals are discussed below, which are not quite as important as relevance but are nevertheless important enough to have a significant impact.

Novelty: Recommender systems are truly helpful when the recommended item is something that the user has not seen in the past. For example, popular movies of a preferred genre would rarely be novel to the user. Repeated recommendation of popular items can also lead to reduction in sales diversity.

Serendipity: A related notion is that of serendipity, wherein the items recommended are somewhat unexpected, and therefore there is a modest element of lucky discovery, as opposed to obvious recommendations. Serendipity is different from novelty in that the recommendations are truly surprising to the user, rather than simply something they did not know about before. It may often be the case that a particular user may only be consuming items of a specific type, although a latent interest in items of other types may exist which the user might themselves find surprising. Unlike novelty, serendipitous methods focus on discovering such recommendations.

Increasing recommendation diversity: Recommender systems typically suggest a list of top-k items. When all these recommended items are very similar, it increases the risk that the user might not like any of these items. On the other hand, when the recommended list contains items of different types, there is a greater chance that the user might like at least one of these items. Diversity has the benefit of ensuring that the user does not get bored by repeated recommendation of similar items [10].

CONCLUSIONS

Recommendation is a suggestion that can help in making good decisions faster. It will help customers and businesses to close transactions faster. As you know by now, recommender systems take data collected on existing user behaviours, and use it to determine what users might also like. By using past behaviour of a large number of people, the taste preferences of an individual or community can be predicted. Recommendation engines are value drivers for Big Data applications. Although hybrid recommender systems have matured as a research area there are still issues related to their accuracy. On the other hand people want to use an intelligent system to assist them in the decision making process in various online environments such as university and commerce domains among others [17, 18]. Furthermore, a recommender system needs to supply as accurate recommendations as possible,

since its job is to substitute a human expert in a particular area. Thus, a Framework for Cloud Based Hybrid Recommender System (FCHRS) for Big Data Mining is propossed.

Finally the recommender systems are highly applicable, and in real world situations, different approaches are combined to form a hybrid recommender system for better results. In addition, the combination of results of two algorithms provides more useful business intelligence. This combination is the new and unexplored area of research. It will be more useful for adding big value to enterprises.

REFERENCES

1. Y. Sowmya, Parallelizing K-Anonymity Algorithm for Privacy Preserving Knowledge Discovery from Big Data, Int. J. Appl. Eng. Res., 11, (2), 2016, 1314-1321.
2. Y. Yengi, S.İ. Omurca, Distributed Recommender Systems with Sentiment Analysis Büyük Veride Tavsiye Sistemlerini Duygu Analizi ile Desteklemek, Eur. J. Sci. Technol., 4, (7), 2016, 51-57.
3. J.P. Verma, Big Data Analysis : Recommendation System with Hadoop Framework, IEEE Int. Conf. Comput. Intell. Commun. Technol., pp. 92-97, 2015.
4. A. Alluhaidan, Recommender System Using Collaborative Filtering Algorithm, 2013.
5. J. Kim, S. Hwang, Big Data Platform of a System Recommendation in Cloud Environment, Int. J. Softw. Eng. Its Appl., 9, (12), 2015, 133-142.
6. B.M.D. Ekstrand, J.T. Riedl, Collaborative Filtering Recommender Systems and Joseph A. Konstan, 4, (2), 2011, 81-173.
7. S.K. Zhuo Zhang, Paul Cuff, Iterative collaborative filtering for recommender systems with sparse data, Princeton University, Department of Electrical Engineering, Princeton, NJ 08544, IEEE Int. Work. Mach. Learn. SIGNAL Process., 2012, pp. 1-6.
8. X. Fang, J. Zhan, Sentiment analysis using product review data, J. Big Data, 2015, pp. 1-14.
9. S. Bagchi, Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout, Procedia Comput. Sci., v. 50, 2015, pp. 229-234.
10. H. Lee, An Introduction to Recommender Systems, Springer Int. Publ. Switz., 2016, pp. 1-29.
11. K.U. Jaseena, J.M. David, Issues, Challenges, and Solutions: Big Data Mining, Comput. Sci. Inf. Technol., 2014, pp. 131-140.12. http://blog.algorithmia.com/introduction-sentiment-analysis-algorithms/
13. https://algorithmia.com/algorithms/nlp/SentimentAnalysis
14. https://www.cloudera.com/documentation/other/tutorial/CDH5/topics/ht_example_4_sentiment_analysis.html
15. http://bigdataknowhow.weebly.com/blog/big-data-recommendation-engines-overview
16. https://mahout.apache.org/users/recommender/userbased-5-minutes.html
17. F. Tomova, Application of the MicroStrategy BI platform to calculate claim ratio for the insurance companies, XI-th international conference „Challenges in higher education and research in the 21st century“, Sozopol, Bulgaria, 2013.
18. F. Tomova, Research on methods of predicting the state of the Peirce-Smith converters for the purposes of predictive maintenance, Journal of Automatics and Informatics, 2, 2014, 14-21.