Journal of Information & Knowledge Management
Vol. 18, No. 1 (2019) 1950009 (21 pages)

DOI: 10.1142/S0219649219500096

A Big Data Recommendation Engine Framework Based on Local Pattern Analytics Strategy for Mining Multi-Sourced Big Data

T. Venkatesan
Department of Computer Science
PRIST University, Thanjavur, Tamilnadu, India
tvntvn09@gmail.com

K. Saravanan
Faculty of Computer Science
PRIST University, Thanjavur, Tamilnadu, India
ks_tnj@yahoo.com

T. Ramkumar
School of Information Technology & Engineering
VIT University, Vellore, Tamilnadu, India
ramkumar.thirunavukarasu@vit.ac.in

Published 21 January 2019

Abstract. Organisations that perform business operations in a multi-sourced big data environment are in imperative need to discover meaningful patterns of interest from their diversified data sources. With the advent of big data technologies such as Hadoop and Spark, commodity hardwares play vital role in the task of data analytics and process the multi-sourced and multi-formatted big data in a reasonable cost and time. Though various data analytic techniques exist in the context of big data, recommendation system is more popular in web-based business applications to suggest suitable products, services, and items to potential customers. In this paper, we put forth a big data recommendation engine framework based on local pattern analytics strategy to explore user preferences and taste for both branch level and central level decisions. The framework encourages the practice of moving computing environment towards the data source location and avoids forceful integration of data. Further it assists decision makers to reap hidden preferences and taste of users from branch data sources for an effective customer campaign. The novelty of the framework has been evaluated in the benchmark dataset, *MovieLens100k* and results clearly confirm the advantages of the proposal.

Keywords: Big data; local pattern analytics; collaborative filtering; mining big data; multi-sourced data.

1. Introduction

Impact of social networking, emergence of Internet-of-Things, usage of ubiquitous devices created avenues for progressive growth of big data that has been characterised by three attributes namely Volume, Velocity and Variety. Though disparateness, huge volume, existence of heterogeneity and velocity are the unique features of big data, these properties also bring significant complexities while performing data intensive analytics (Zhou *et al.*, 2017). To analyse the generated big data and identify the real insight in it, distributed computing environment with massively parallel processing architecture has been identified as an efficient solution. With the existing architecture of centralised data warehouse, data analyst must spend significant effort and time to aggregate data from multiple sources (Ramkumar *et al.*, 2013). It leads to several drawbacks such as (i) Requirement of huge investment for purchasing hardware and software entities; (ii) Challenging data ingestion task since multiple data sources share un-common formats; (iii) Hard to analyse local trends and deviations found in individual data sources since forceful data integration took place; (iv) Due to privacy laws and legal issues, some of the data owners are shown their dis-interest in forwarding their branch level data.

In such circumstances, the strategy of *local pattern analytics* has been put forward as an efficient alternative strategy for dealing data sources located in multiple places (Zhang *et al.*, 2004b). *A local pattern analytics strategy is one which efficiently identify and forward patterns such as frequent objects, supervised models, similar groups and even potential recommendations/rating of user community from individual data sources and the source of data may be structured, un-structured and even semi-structured one* (Ramkumar *et al.*, 2014a). The process of local pattern analytics has been architecturally shown in Fig. 1. Let us consider "N" big data sources namely "BD_1", "BD_2", ..., "BD_N" with respect to "N" branches of an interstate organisation.

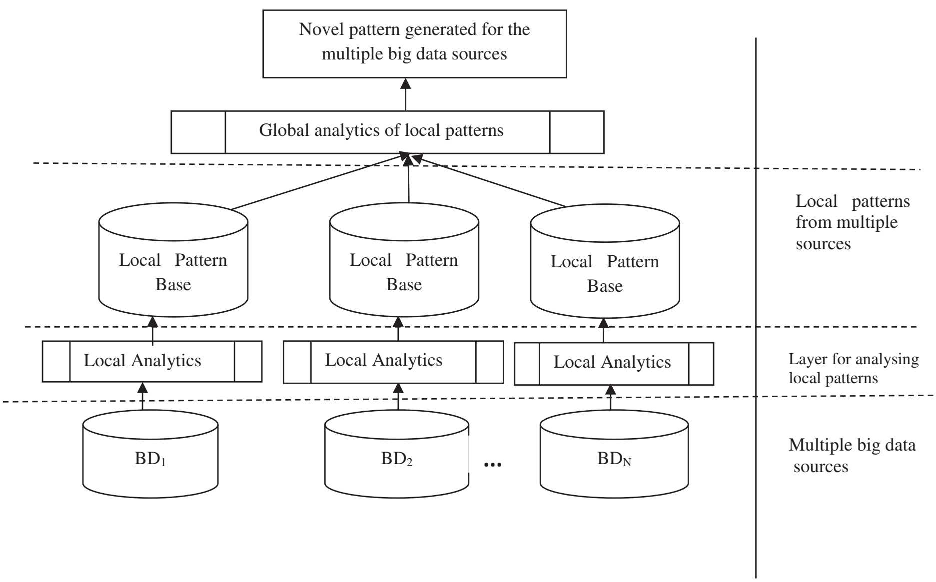

Fig. 1. Local pattern analytics strategy.

To reap novel patterns from these individual data sources, appropriate analytical and mining techniques can be applied to obtain interesting pattern base with respect to individual sources. These pattern bases are further forwarded for performing global analytics. Hence, the strategy of local pattern analytics effectively analyse, evaluate and synthesise the patterns of interest without obliterating the individual/special features found in the big data sources. The strategy of local pattern has been initially put forth by Zhang *et al.* (2003) and the major differences centralised mining and local pattern analytics are deliberately discussed in their work.

By adopting the strategy of local pattern analytics, the following advantages have been evidenced by the branch and central level decision makers (Zhang and Zaki, 2006). (i) *In-place strategy*: Feasible way to generate pattern base when huge volume of data are distributed among various sites or branches. (ii) *Two-Level decision*: Provision for two-level decision such as global decision for central organisation and local decision for individual branches. (iii) *Less-complexity*: Only pattern bases are forwarded for the task of data analytics and no need to integrate the sources of big data.

The influence of big data analytics has been greatly recognised among data science community in domains such as healthcare, telecommunication, insurance and web information retrieval (Tsai *et al.*, 2015; Ahamed and Ramkumar, 2016; Palanisamy and Thirunavukarasu, 2017) With the advent of open source software platforms and languages, organisations are moving towards big data technologies for solving their complex computational needs. As a pioneering open source software framework, Hadoop offers a distributed file system (HDFS) and data processing engine (Map-Reduce programming model) to handle extremely high volume of data in any structure. It emerged as flexible, scalable and highly available architecture for large scale data processing on network of commodity hardware (commercially available hardware) with a reasonable cost and time. Though HDFS and Map-Reduce daemons are core components of Hadoop, the complexities involved in writing functional style programming of Map-Reduce in terms of Key–Value pair prompt the big data analysts to develop various eco-system components for hiding the complexities of Map-Reduce programming model. Some of the well-known big data eco-system components are Hive (for performing SQL operations), Pig (to perform data pre-processing operations), Sqoop (for importing data from relational models) and Mahout (for performing machine learning algorithms).

As a distributed machine learning library, Mahout performs machine learning tasks such as classification, clustering, association rule mining and recommender system. In the context of big data analytics, recommender system has been emerged as a decision supportive framework for both service providers and service consumers (Isinkaye *et al.*, 2015) and help organisation to identify suitable consumers and services by personalising user preferences. Especially for organisations that perform business operations in multiple locations across different regions, identification of user preferences/tastes with respect to branch data sources would play an important role. Our intention in this paper is to implement the advantages of local pattern analytics strategy for analysing sources of big data by developing a novel big data processing framework. The proposed framework processes the big data through a recommendation engine for identifying local and global features of disparate big data sources. The framework would be an ideal solution for processing huge volume of structured, semi-structured and un-structured data sources in big data platforms such as Hadoop and Spark with the aid of appropriate machine learning libraries.

The rest of the paper has been structured as follows: Sec. 2 of the paper elaborates the promising research outcomes found both in local pattern analytics strategy and big data analytics. The proposed big data framework has been introduced in Sec. 3 and the need for such framework is also justified. In Sec. 4, the novelty of the proposal has been validated with the aid of benchmark dataset and the results are elaborately studied. Section 5 concludes the paper with a scope for future work.

2. Related Research Work

Since our proposed work aims to construct a big data framework on the basis of local pattern analytics strategy, this section discusses various salient research attempts in these two broad areas along with their outcomes.

2.1. *Research attempts based on local pattern analytics strategy*

The main objective of the local pattern analytic strategy is to perform the analytical operations over the underlying distributed data sources without moving the data into the centralised repository. As a result, the strategy effectively reaps regional patterns or features from the branch level data sources and forwards those patterns to the central level operations for further processing. A local pattern would be a frequent item-set, an association rule, a casual dependency or some form of interesting information (Zhang *et al.*, 2004b). Business organisations that perform voluminous transactions across multiple branches are in imperative need for arriving two levels of decisions namely global and local decisions. Following this logic, Zhang *et al.* (2003) classified the local patterns into high-vote patterns, suggested patterns and exceptional patterns. When more number of local patterns forwarded from branch data sources, a synthesising model is necessary to gather global patterns from the forwarded local patterns. Wu and Zhang (2003) advocated a weighting model for synthesising high-frequency association rules from different data sources. An association rule is called as high-frequency rule if it is supported or voted by large number of participatory data sources. The weight of participating data source in turn calculated on the basis of number of high-frequency rules voted by it.

Zhang *et al.* (2004a) attempted an approach for synthesising exceptional patterns globally from multiple data sources. They concluded that exceptional patterns reflect the individuality of branches within an interstate organisation and they developed an algorithm for identifying global-exceptional patterns from multiple data sources. Their attempt has been considered as post-processing work after mining multiple, relevant data sources. Nedunchezhian and Anbumani (2006) focussed on the problem of data source selection in mining multi-database applications. They have suggested threshold value for identifying candidate data sources to participate in the pattern synthesising process. Adhikari and Rao (2008) extended the local pattern analytics model by introducing the notion of heavy association rules. They claimed that heavy association rules would be more interesting than high-frequent association rules and such rules may not be shared by all the participating data sources. They proposed an algorithm for synthesising heavy association rules from multiple data sources and reported whether a heavy association rule is high-frequent or exceptional among participatory data sources.

Ramkumar and Srinivasan (2008) proposed a weighting model for synthesising high-frequent association rules from participating data sources. The weight of data source has been calculated on the basis of data source population (number of transaction in the database). Their goal in synthesising global patterns is that the synthesised support and confidence should be nearly same if all the data sources are integrated and central mining has been done. To incorporate the importance of weak association rules which fail to satisfy the minimum threshold value, the notion of correction factor has also been introduced by them (Ramkumar and Srinivasan, 2009). They further added that suitable correction factor has to be chosen by the domain expert on the basis of distribution of data and domain knowledge. With the inclusion of correction factor, they have shown improved synthesised results.

Adhikari *et al.* (2009) proposed a synthesising model for mining global patterns from time stamped transactional databases on the basis of degree of stability of an item. They have emphasised the need for finding variation of sales of an item over time in a multi-database scenario. Zhang *et al.* (2009) employed the concept of Customer Lifetime Value (CLV) for obtaining local patterns from individual data sources. The attributes of customer namely customer-id, expenditure amount, time of retention are used to compute the CLV of a customer. By using the method namely Kernel estimation, global patterns are synthesised from local patterns. A multi-level rule synthesis model has been advocated by Ramkumar and Srinivasan (2010) to extend their earlier weighting model. In their work, they introduced two rule selection measures namely *effective vote rate* and *nominal vote rate*. By using these measures, local patterns have been synthesised into global, sub-global and local patterns. He *et al.* (2010) proposed a synthesising model for data sources which are irrelevant with each other. They emphasised that simple rule synthesising models without a detailed understanding of underlying data sources are not adequate for reaping meaningful patterns. Accordingly, for databases of different features, they have performed clustering at item level first and applied rule synthesising method later.

In an inter-state business organisation, many important decisions are based on the sales of certain important specific item-sets. Adhikari *et al.* (2011) represented those items as *select items* and a measure for identifying association among *select items* has been proposed. Further, they designed an algorithm on the basis of proposed measure for grouping *select items* from multiple databases. Adhikari (2013) attributed the importance of clustering local frequency items from multiple databases. They have proposed a clustering technique that clusters local frequency item at higher level by capturing association among items using appropriate relevant measure. The notion of negative association rule in the context of multi-database system has been put forth by Ramkumar *et al.* (2014b). They emphasised that negations among item-sets are also important for identifying conflict of interest in a typical multi-database application. A generic synthesising model to evaluate the interestingness of pattern on the basis of desired interestingness measures has been proposed by Ramkumar *et al.* (2014a). They have employed 17 interesting rule evaluation measures and explored the significance of synthesised patterns. The salient features of above research attempts and their outcomes are summarised in Table 1.

The following section reviews prominent research attempts on various analytical perspectives for managing big data sources.

Table 1. Summary of research efforts in local pattern analytics strategy.

Paper studied	Focussed problem	Analytical strategy	Research outcome
Zhang *et al.* (2003)	Limitation of centralised mining	In-place analytical strategy	Identification of new kinds of patterns in a multi-database environment
Wu and Zhang (2003)	Discovery of global patterns from multi-database system	Pattern synthesising model	Data source weighting model for global pattern discovery
Zhang *et al.* (2004a)	Discovery of new kinds of patterns in multi-database system	Algorithm for identifying exceptional patterns	Weighting model for identifying globally exceptional patterns
Nedunchezhian and Anbumani (2006)	Data source selection for multi-database application	Selection procedure for identifying candidate data sources	Two-level synthesising model for data source selection and global pattern discovery
Adhikari and Rao (2008)	Discovery of new kinds of patterns in multi-database system	Pattern synthesising model for heavy association rules	Strategies for identifying whether heavy association rule is high-frequent or exceptional among participatory data sources.
Ramkumar and Srinivasan (2008)	Discovery of global association rules	Pattern synthesising model	Transaction-population-based data source weighting model
Ramkumar and Srinivasan (2009)	Inclusion of weaker patterns in synthesising process	Choosing suitable correction factor in pattern discovery	Improved synthesised results with the inclusion of correction factor

Table 1. (*Continued*)

Paper studied	Focussed problem	Analytical strategy	Research outcome
Adhikari *et al.* (2009)	Stability of patterns in multi-database applications	Synthesising model for pattern stability	Pattern synthesising model for time-stamped transactional databases
Zhang *et al.* (2009)	Inclusion of customer specific attributes in pattern discovery	Synthesising model based on CLV	Improved synthesised results using CLV
Ramkumar and Srinivasan (2010)	Post-processing of identified patterns	Multi-level pattern synthesis model	Rule evaluation measures for classifying the patterns into global, sub-global and local features
He *et al.* (2010)	Clustering of data sources	Synthesising model for data sources of irrelevant nature	Two-Level synthesizing model with improved result
Adhikari *et al.* (2011)	Significance of specific item-sets in multi-database application	Algorithm for grouping selective items	Measure for identifying association among selective items
Adhikari (2013)	Clustering of patterns	Clustering technique for grouping local frequent item-sets	Relevant measure for grouping frequent items-sets in multi-database system
Ramkumar *et al.* (2014b)	Discovery of new kinds of patterns in multi-database system	Synthesising model for negative association rules	Discovery of global negative patterns
Ramkumar *et al.* (2014a)	Evaluation of synthesised patterns	A generic framework for synthesising patterns	Synthesising model for desired interestingness measures

2.2. *Research attempts on analytical approaches for managing big data sources*

Gillick *et al.* (2006) advocated the HDFS and Map-Reduce programming model for the implementation of standard machine learning algorithms such as single-pass learning, iterative learning, and distance-based learning. By doing so, they have investigated efficient strategies for sharing data among computing nodes. Further they concluded that Map-Reduce-based machine learning algorithms hold good for parallel learning.

Ranger *et al.* (2007) attempted the evaluation of Map-Reduce programming model for multi-core and multi-processor system by developing an application programming interface called Phoenix. The interface effectively performs parallel execution of data mining algorithms such as k-means clustering, principal component analysis and linear regression based on Map-Reduce programming model.

The problem of parallelising data mining and machine learning algorithms for handling big data sources has been tackled by Luo *et al.* (2012) since the task of parallelisation is non-trivial. They proposed a strategy to parallelise series of data mining algorithms such as support vector machine and linear regression models using Map-Reduce programming models.

One of the most challenging tasks in big data applications is to efficiently handle the large volume of data towards obtaining actionable insights. Hence the unpredicted growth of big data requires novel analytical strategies based on descriptive, predictive and prescriptive models. Wu *et al.* (2014) advocated HACE theorem to bring the analytical importance of big data. They defined big data as Heterogeneous, Autonomous repositories with distributed and decentralised control for seeking Complex and Evolving relationships. It shows that the centralised analytical strategy would not be a feasible solution for big data analytics. In addition they put forth a three-tier big data framework that includes big data mining platform, big data semantics with application knowledge, and big data mining algorithms.

Meng *et al.* (2014) proposed a scalable machine learning recommendation framework called KASR (Keyword-Aware Service Recommendation) for performing collaborative recommendation. In their approach, keywords are used to indicate user's preferences, and a user-based collaborative filtering algorithm is adopted to generate appropriate recommendations. To improve its scalability and efficiency in big data environment, they implemented their framework on Hadoop using Map-Reduce parallel processing paradigm.

A cloud-based commercial recommendation system has been put forth by Pereira *et al.* (2014). The proposed framework provides various value added features such as computation of online similarity and scale-up, scale-down of number of items and users. The state of the work has been implemented on video recommendation domain. The focus on improving the efficiency of recommendation algorithm in the context of big data has been carried out by Zhang *et al.* (2016). They have conducted experiments for the reduction of computation time and evaluated in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) metrics which are most representative and effective ones.

Elahi *et al.* (2016) surveyed the role of active learning in collaborative filtering recommendations. As a sub-field of machine learning, active learning provides several machine learning tasks when significant amount of high-quality data are not available. The strategy of active learning adopts criteria for obtaining data that better reflects user preferences. Bu *et al.* (2016) introduced a new Multiclass Co-Clustering (MCoC) model, which captures relations among user-to-item, user-to-user, and item-to-item simultaneously. Then they combined traditional collaborative filtering algorithms with sub-groups for improving their top-N recommendation performance.

Choi *et al.* (2016) tackled the problem of identifying dynamic user preferences for recommendation of items. They proposed a recommendation procedure using facial expression of users for recommending online videos when past ratings of users and purchase records are not available. By using facial expression as basic element in recommending items for new users, they have addressed solution for cold start problem also. The impact of recommendation system in biological domain has been envisioned by Hu *et al.* (2017). They have applied collaborative filtering in biological data for gene prediction and developed an algorithm namely "Top-N Gene-Based Collaborative Filtering" to explore the gene interest (Gi) and recommend the genes for individual patients. The proposed recommendation algorithm identifies six major genes that cause liver cancer and help physicians to provide smarter, customised care for cancer patients.

Geuens *et al.* (2018) proposed a decision supportive framework for e-commerce companies to generate best collaborative recommendations based on the purchase data. By providing the customer's shopping data source as sparse dataset with specific input characteristics, the framework offers an indication about which algorithm is more suitable for performing recommendation through evaluation metrics such as accuracy, diversity and time complexity. Shams and Haratizadeh (2018) formulised the user-based, representative-based and preference-based recommendation approaches into a graph-based collaborative filtering framework which uses *Tripartite* Preference Graph (TPG) that contains set of users, pair-wise preferences and items to directly estimate the interest of user over an item.

Table 2. Summary of analytical approaches for managing big data sources.

Paper studied	Focussed problem	Analytical strategy	Research outcome
Gillick *et al.* (2006)	Distributed load sharing among computing nodes	Map-Reduce programming model	Distributed machine algorithms for analysing structured big data sources
Ranger *et al.* (2007)	Multi-core systems for processing big data sources	Map-Reduce programming model	Development of novel application programming interface based on Map-Reduce programming model
Luo *et al.* (2012)	Distributed data mining systems	Map-Reduce programming model	Parallelisation of specific data mining algorithms
Wu *et al.* (2014)	Big data complexities	Data analytics models	A novel theorem for big data and a three-tier big data processing framework
Meng *et al.* (2014)	Recommendation system	Map-Reduce programming model	Scalable machine learning recommendation framework
Pereira *et al.* (2014)	Recommendation system	Online similarity computation	Cloud-based scalable recommendation system

Table 2. (*Continued*)

Paper studied	Focussed problem	Analytical strategy	Research outcome
Zhang *et al.* (2016)	Recommendation system	Reduction of computation time	Strategies for improving the efficiency of recommendation algorithm
Elahi *et al.* (2016)	Quality of big data sources	Active learning	Criteria for obtaining better user preferences
Bu *et al.* (2016)	Relationship among recommendation objects	Clustering approach	A clustering model for recommendation systems
Choi *et al.* (2016)	Identifying dynamic preferences of users	Collaborative filtering with facial expressions	Rating prediction system for video recommendation
Hu *et al.* (2017)	Recommendation system in biological domain	Collaborative filtering	Prediction of genes for cancer disease
Geuens *et al.* (2018)	Recommendation system accuracy	Evaluation of recommendation algorithms	Decision supportive framework for e-commerce-based recommendation systems
Shams and Haratizadeh (2018)	Relationship among recommendation objects	Hybrid approach for recommendation system	Graph-based collaborative filtering framework
Hwangbo *et al.* (2018)	Dynamic preferences of users	Item-based collaborative filtering with the inclusion of domain characteristics	Recommender system for fashion retail industry
Rezaeimehr *et al.* (2018)	Dynamic preferences of users	Rule mining in recommendation system	Recommender system based on temporal factors

To capture the changing preferences of users about seasonal fashion products, Hwangbo *et al.* (2018) proposed a new recommendation system called as "K-RecSys". The proposed system expands the item-based recommendation algorithm for reflecting the domain characteristics of fashion retail industry. They have considered the user preferences through online click stream data and offline purchase data. Further they proposed a preference decay function to reflect changes in preferences over time and finally the system recommends substitute and complementary products by using product category information. Rezaeimehr *et al.* (2018) considered the temporal effects in historical ratings and also the similarity values between the users to propose a novel recommendation algorithm. A reliability measure is also proposed by them to identify the relevant and reliable recommendations. Finally a rule mining approach has been implemented in the predicted recommendations to identify the temporal effects on historical ratings. The summary of above research attempts along with their significant research outcomes is presented in Table 2.

3. Proposed Big Data Recommendation Engine Framework

In our proposal, we have employed local pattern analytics strategy in the context of big data analytics for analysing multiple disparate big data sources to reap meaningful features for arriving global and local level decisions. As a pattern discovery technique, recommender systems have attracted significant attention due to the increasing number of e-commerce users. Users offer rating on purchased items and recommendation system uses this information to predict their preferences for the yet unseen items and subsequently recommend items with predicted rating. In our framework, an item-based collaborative filtering approach has been applied as a recommendation technique to analyse the local and global preferences of users over the participatory big data sources.

The proposed big data recommendation engine framework is shown in Fig. 2. There are "N" commodity hardware machines namely CH_1, CH_2, ... , CH_N for handling big data sources BD_1, $\mathrm{BD}_2, \ldots,$ BD_N, respectively. A standard Hadoop cluster has been deployed on commodity hardware to process those big data sources. The daemons of HDFS namely Name node and Data node are responsible for managing and storing distributed sources of big data in the Hadoop environment. The two daemons namely Job-tracker and Task-tracker of Map-Reduce programing model are responsible for submission and execution of jobs requested by the user and functions namely Mapper and Reducer execute the respective jobs in terms of key-value pairs. On the top of this core layer, Mahout has been installed as an eco-system

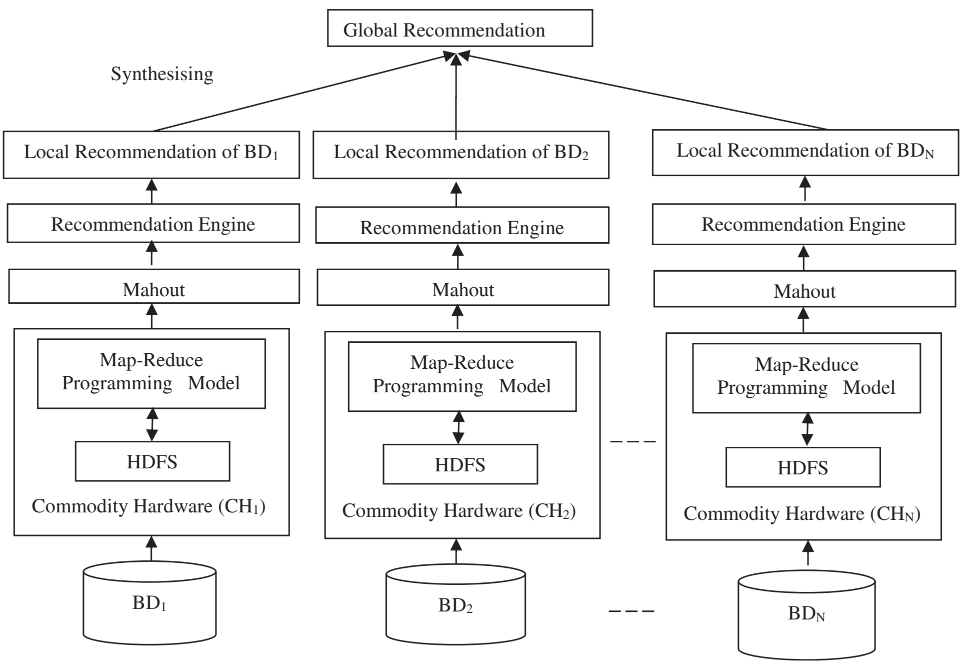

Fig. 2. Proposed big data processing framework.

component to perform various machine learning techniques. In our framework, mahout performs item-based collaborative recommendation algorithm on each Hadoop cluster to predict the user rating for an item by computing the similarity among items. The predicted recommendation of items for a user can be considered as local recommendation of respective data sources. Let $P(D, U, I : R)$ is the local pattern or recommendation generated from a data source "D" along with the predicted rating "R" on item "I" for a user "U". The generated local recommendations reap the branch-level user preferences and further forwarded to central head for performing global analytics.

It has been noted that the architecture of the proposed framework closely matches with the processing architecture of local pattern analytics strategy (Zhang *et al.*, 2003). From both Figs. 1 and 2, it has been further observed that one can apply local pattern analytics strategy as an intuitive framework for mining big data sources with the advent of HDFS and Map-Reduce programming model.

The proposed framework can be extended for other big data platform such as Apache Spark also. Apache Spark is an open source parallel processing framework for running large-scale data analytics applications across cluster of computers. It handles both batch and real-time data processing workloads. The proposed framework can be easily portable for a Spark distributed environment by replacing the Map-Reduce programing model with a Resilient Distributed Dataset (RDD) programming and Mahout can be replaced by MLlib (Apache Spark's scalable machine learning library). Hence, the outcome of proposed framework delivers designated pattern of interest based on the underlying machine learning algorithms adopted.

By adopting the proposed framework for processing multiple big data sources, the following benefits are resulted: (i) *Avoiding data ingestion*: Since big data sources fall under different data formats, ingestion of those data sources into a centralised location and formation of unified schema structure would always be a tedious task. With the advent of proposed framework, data movement has been effectively obliterated. (ii) *Effective utilisation of commodity hardware*: The proposed framework forms a massively parallel processing architecture by using commodity hardware machines. They are commercially available hardware machines (laptop and personal computers) and generally compatible with other such devices. Hence the proposed framework does not require high-end server machines for processing big data sources. As a result, infrastructure cost can be considerably reduced. Also the computing environment can easily be moved towards the location of data. (iii) *Effective handling of stream oriented data*: To perform real-time analytical processing of stream data, incremental mining techniques are highly essential.In such circumstances, the traditional architecture of centralised strategy would not be an effective solution since series of data staging activities are involved while performing data movement. With the advent of big data eco-system components, the proposed architecture holds the capability to handle stream oriented data in an effective manner. (iv) *Handling data of veracity nature*: Veracity is one of the important characteristics of big data which bounds with an idea to investigate data of uncertainty nature. Factors such as data in-completeness, in-consistencies, ambiguities, latency and approximation are the main causes for data veracity. To handle such data with *doubtless*, the proposed architecture would be an ideal solution while comparing the centralised analytical strategy since the latter swipes and merges the data sources. (v) *Multi-level discovery of patterns*: Since our proposed framework performs pattern discovery at two levels of abstraction namely local and global levels, novel pattern synthesising strategies would be resulted by adopting the proposed framework.

3.1. *Item-based collaborative filtering*

Item-based collaborative filtering is the type of recommendation system proposed by Sarwar *et al.* (2001) to compute the similarity between items and then predict the rating on item. It uses user/item matrix where rows represent users and columns represent items. To illustrate the working procedure of item-based collaborative filtering, a user/item matrix has been constructed as an example with four users and three items (Table 3). The range of rating is between 1 (lowest rating) and 5 (highest rating).

In the above example, to infer the predicted rating of "User 1" on "Item 2", item-based collaborative filtering approach performs similarity computation followed by prediction of rating.

Item-similarity computation: The basic idea in similarity computation between two items "i" and "j" is to first identify users who have rated both of these items and then apply a similarity computation technique to determine the similarity, $S_{i,j}$. There are many measures available such as Euclidean Distance, Cosine and Pearson-Correlation for computing similarity among items. We have chosen Pearson-Correlation, because in terms of recommendation time, it performs far better than other similarity measures in the Mahout environment:

$$\text{Pearson Similarity}(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_j)^2}}, \quad (1)$$

Table 3. User/item matrix for an example data.

	Item 1	Item 2	Item 3
User 1	2	?	3
User 2	5	2	2
User 3	3	3	1
User 4	3	2	2

where

U — domain of all users,
u — a user,
i, j — some items,
$R_{u,i}$ — Rating of user, "u" on item, "i",
$\bar{R}_i$ — Mean rating value for item, "i",
$R_{u,j}$ — Rating of user, "u" on item, "j",
$\bar{R}_j$ — Mean rating value for item, "j".

By using Eq. (1), similarities among the items are calculated for the example data

$$\text{Similarity (item2, item1)}$$
$$= \frac{(2-2.3)*(5-3.7)+(3-2.3)*(3-3.7)+(2-2.3)*(3-3.7)}{\sqrt{(2-2.3)^2+(3-2.3)^2+(2-2.3)^2}*\sqrt{(5-3.7)^2+(3-3.7)^2+(3-3.7)^2}}$$
$$= \frac{-0.67}{1.3374} = -0.5009,$$

$$\text{Similarity (item2, item3)}$$
$$= \frac{(2-2.3)*(2-1.7)+(3-2.3)*(1-1.7)+(2-2.3)*(2-1.7)}{\sqrt{(2-2.3)^2+(3-2.3)^2+(2-2.3)^2}*\sqrt{(2-1.7)^2+(1-1.7)^2+(2-1.7)^2}}$$
$$= \frac{-0.67}{0.67} = -1.$$

Prediction of rating: To predict the rating on item for a target user, we applied weighted sum method. It computes the prediction on an item, "i" for a user, "u" by computing sum of the ratings given by the user on the item similar to item "i" and each rating is weighted by the corresponding similarity, $S_{i,j}$ between items "i" and "j". The prediction $P_{u,i}$ is computed as

$$P_{u,i} = \frac{\sum_{\text{all similar items},N}(S_{i,N} * R_{u,N})}{\sum_{\text{all similar items},N}(|S_{i,N}|)}. \tag{2}$$

By applying Eq. (2),

$$P_{(\text{User1,Item2})} = \frac{(-0.5009)*2+(-1)*3}{(-0.5009)+(-1)}$$
$$= \frac{-4.0018}{-1.5009} = 2.66.$$

Hence the predicted rating on "Item2" by "User1" is 2.66.

Though regression-based prediction technique is there, it uses approximation ratings instead of directly using the ratings of similar items. Hence we adopt weighted sum method to predict the rating.

4. Experimental Evaluation

This section elaborates the experimental investigation performed to validate the effectiveness of proposed recommendation engine framework.

4.1. *Experimental setup and dataset description*

The experimental setup has been instituted by using Hadoop version 1.2.1 along with eco-system component Mahout 11.00. As an open source machine learning library, Mahout offers wide range of machine learning features to support distributed processing of large datasets across cluster of nodes using HDFS. To perform item-based collaborative filtering on Hadoop cluster, the benchmark dataset, *MovieLens100k* (http://grouplens.org/datasets/movielens/) has been used in our experiment. The dataset contains 100,000 transactions applied to 1682 movies by 943 users of the online movie recommender service. The data are contained in three files namely *movies.dat*, *ratings.dat* and *tags.dat*. The file *ratings.dat* contains at least three columns: the user ID, the item ID, and the rating value. Each user has at least rated 20 movies and the range of ratings is from 1 (lowest) to 5 (highest). To implement the proposed framework, the entire dataset of 1,00,000 transactions has been divided into 50,000, 30,000 and 20,000 and designated with respective sites namely, "Site1", "Site2" and "Site3". A multi-node Hadoop cluster architecture with three nodes has been advocated for processing these three data sources and each node has a specification of quad-core 2–2.5 GHz CPU processor, 8 GB RAM, Four 1TB SATA disks and a Gigabit Ethernet network card. The experimental settings are summarised in Table 4.

To perform recommendation engine on each cluster, Mahout has been installed on the top of the Hadoop and item-based collaborative filtering algorithm has been executed in each of the node. In our experimental work, each movie has been considered as an item and the movie ratings are considered as ratings for items. The algorithm computes similarity by using Pearson correlation measure and performs prediction on the basis of weighted sum technique.

4.2. *Results and discussion*

In "Site1", 762 users have rated the total amount of 1590 items and in "Site2", a total number of 942 users have rated 1478 items. Similarly in "Site3", a total number

Table 4. Summary of experimental settings.

Processor	Quad Core 2.5 GHz processor
RAM	8 GB 1067 MHz DDR3
Operating systems	Ubuntu 16.0.2
JVM	JRE 1.6.0_65
Hadoop	Hadoop 1.2.1
Number of clusters	3
Number of local data sources	3
Mahout	Apache Mahout 11.00
Mahout API	PearsonCorrelationSimilarity
Data File	Movie lens data of 27.2 MB
User–item preferences rating scales	1 to 5

of 927 users have purchased and rated 1407 items. To take an effective branch level decision and to know about the customer's taste at individual site, it is appropriate to infer predicted recommendation of user for the non-purchased items also. For example, the "User1" in "Site1" has purchased 212 items alone and for rest of 1378 items, his target rating would be a quite interesting one for the branch level decisions. To infer the recommended ratings of items at branch level, we have executed item-based collaborative filtering algorithm on all three nodes and inferred the taste of the users. We have tabulated first 10 users and their top 10 recommended items along with the predicted rating values for the non-purchased items found in all three sites (Table 5).

Table 5. Top 10 items recommended for first 10 users along with predicted ratings.

Hadoop cluster (Sites)	User ID	{Item id:rating}
Site1with a preference population of 50,000	1	[{16:5}, {19:5}, {7:5}, {13:5}, {17:5}, {18:5}, {4:5}, {5:5}, {11:5}, {20:5}]
	2	[{1089:5}, {566:5}, {327:5}, {900:5}, {363:5}, {496:5}, {168:5}, {144:5}, {176:5}, {1099:5}]
	3	[{267:5}, {763:5}, {850:5}, {410:5}, {1123:5}, {128:5}, {619:5}, {669:5}, {551:4.8}, {800:4.7}]
	4	[{31:5}, {20:5}, {1062:5}, {27:5}, {1056:5}, {19:5}, {16:5}, {1:5}, {1045:5}, {1098:5}]
	5	[{12:5}, {15:5}, {6:5}.{11:5}, {13:5}, {14:5}, {3:5}, {4:5}, {7:5}, {16:5}]
	6	[{8:5}, {11:5}, {4:5}, {7:5}, {9:5}, {10:5}, {1:5}, {2:5}, {6:5}, {12:5}]
	7	[{6:5}, {11:5}, {3:5}, {5:5}, {9:5}, {10:5}, {1:5}, {2:5}, {4:5}, {12:5}]
	8	[{8:5}, {13:5}, {4:5}, {7:5}, {11:5}, {12:5}, {1039:5}, {1:5}, {5:5}, {1056:5}]
	9	[{316:5}, {304:5}, {309:5}, {300:5}, {24:5}, {1146:5}, {1267:5}, {1258:5}, {173:5}, {472:5}]
	10	[{6:5}, {9:5}, {3:5}, {5:5}, {7:5}, {8:5}, {1:5}, {2:5}, {4:5}, {11:5}]
Site2 with a preference population of 30,000	1	[{5:5}, {9:5}, {2:5}, {4:5}, {7:5}, {8:5}, {1039:5}, {1:5}, {1042:5}, {12:5}]
	2	[{909:5}, {360:5}, {1463:5}, {651:5}, {889:5}, {882:5}, {1386:5}, {86:5}, {15:5}, {387:5}]
	3	[{1160:5}, {1463:5}, {882:5}, {303:5}, {313:5}, {305:5}, {1423:5}, {327:5}, {293:5}, {361:5}]
	4	[{1405:5}, {305:5}, {1127:5}, {811:5}, {292:5}, {303:5}, {270:5}, {1388:5}, {286:5}, {1137:5}]
	5	[{611:5}, {517:5}, {490:5}, {267:5}, {70:5}, {467:5}, {166:5}, {479:5}, {114:5}, {1451:5}]
	6	[{549:5}, {419:5}, {517:5}, {173:5}, {72:5}, {614:5}, {188:5}, {1168:4}, {52:4}, {479:4}]
	7	[{7:5}, {13:5}, {4:5}, {6:5}, {8:5}, {10:5}, {1:5}, {2:5}, {5:5}, {17:5}]
	8	[{220:5}, {176:5}, {1228:4}, {686:4}, {197:4}, {449:4}, {642:4}, {797:4}, {42:4}, {459:4}]

Table 5. (*Continued*)

Hadoop cluster (Sites)	User ID	{Item id:rating}
	9	[{447:5}, {576:5}, {578:5}, {423:5}, {566:5}, {568:5}, {411:5}, {417:5}, {659:5}, {233:5}]
	10	[{566:5}, {14:5}, {739:5}, {1103:5}, {10:5}, {570:5}, {1115:5}, {2:5}, {561:5}, {572:5}]
Site2 with a preference population of 20,000	1	[{32:5}, {12:5}, {17:5}, {22:5}, {9:5}, {568:5}, {159:5}, {15:5}, {561:5}, {514:5}]
	2	[{412:5}, {134:5}, {99:5}, {946:5}, {1161:5}, {567:5}, {332:5}, {324:5}, {1199:5}, {147:5}]
	3	[{755:4}, {194:4}, {357:3}, {1009:2}, {527:2}, {196:2}, {216:2}, {875:1}, {288:1}, {654:1}]
	4	[{285:5}, {568:5}, {432:4}, {301:4}, {1063:4.}, {502:4}, {98:4}, {26:4}, {217:4}, {111:4}]
	5	[{151:5}, {274:5}, {45:5}, {1110:5}, {141:5}, {96:5}, {15:5}, {9:5}, {61:5}, {340:5}]
	6	[{12:5}, {23:5}, {7:5}, {11:5}, {1060:5}, {22:5}, {189:5}, {1:5}, {10:5}, {24:5}]
	7	[{451:4}, {449:4}, {694:4}, {834:4}, {311:4}, {576:4}, {1099:4}, {1375:4}, {559:4}, {195:4}]
	8	[{492:5}, {89:5}, {479:5}, {1091:5}, {419:5}, {485:5}, {507:4}, {436:4}, {300:4}, {951:4}]
	9	[{168:4}, {480:4}, {402:4}, {163:4}, {689:4}, {26:4}, {330:4}, {603:4}, {285:4}, {544:4}]
	10	[{177:5}, {298:5}, {96:5}, {66:5}, {53:5}, {281:5}, {47:5}, {4:5}, {1048:5}, {323:5}]

From the result, it has been observed that the proposed framework efficiently identifies certain noteworthy items even which have not been purchased by the user. For example, in "Site1", the "User1" has rated "item 155" with a rating level of 2. Though the "item 16" has not been purchased by the same user "User1", his predicted rating is 5. Such kind of interestingness leads effective decision making strategies at branch level.

To identify the potential items at global level, items which have been predicted with higher ratings from the participatory sites are focussed and those items are deemed as global items and shown in Table 6. Though these items have not been purchased in individual sites, they emerged as items with higher predicted rating from all three participatory sites.

Identifying such kind of items and their rating levels would be more useful for strategic level decisions and to know about the hidden preferences of users. The proposed framework efficiently identifies such kind of interesting patterns in a big data scenario. Also, it predicts the recommended ratings for the non-purchased items found in the partcipatory sites and are shown in Tables 7 and 8.

Even though global recommended items which are supported in all three participating sites are important for strategic decision, the recommended ratings for the

Table 6. Globally recommended items for all three sites.

Participating sites	Recommended items
($S1, S2, S3$)	16, 19, 7, 13, 17, 4, 11, 20, 1089, 566, 327, 900, 363, 496, 168, 144, 176, 1099, 410, 128, 619, 34, 1062, 27, 1, 1045, 1698, 12, 15, 14, 8, 9, 10, 1039, 316, 304, 300, 24, 173, 472, 1042, 909, 360, 651, 882, 1386, 86, 387, 1160, 303, 313, 305, 293, 361, 811, 292, 270, 286, 611, 517, 70, 467, 166, 114, 549, 419, 173, 72, 614, 188, 52, 220, 1228, 686, 197, 449, 642, 42, 447, 576, 578, 423, 568, 411, 417, 659, 233, 739, 570, 1115, 561, 572, 32, 22, 159, 514, 134, 99, 946, 1161, 567, 332, 147, 755, 194, 357, 527, 196, 216, 288, 654, 255, 432, 301, 1063, 502, 98, 26, 217, 111, 151, 274, 45, 141, 96, 340, 23, 1060, 189, 22, 451, 194, 311, 1375, 559, 195, 492, 89, 1091, 485, 507, 436, 480, 402, 163, 684, 603, 298, 53, 281, 47, 1048, 323,

Table 7. Recommended items with higher predicted ratings for any two sites.

Participate site	Recommended items
($S1, S2$)	18,56,267,1056,6,3,1267,1258,11271388,490,267,1168
($S1, S3$)	763,669,800,1161,1009
($S2, S3$)	6,479,797,324,1199,875,1110,61,834,951,177,544

Table 8. Recommended items with higher predicted ratings for a single site.

Participate site	Recommended items
$S1$	551,309,1146,459,1103
$S2$	1463,889,1423,1405,1137,1451,419,459,1103
$S3$	412

non-purchased items found in any two sites or from individual site are also playing significant role and those recommendations are deemed as *sub-global* and *local* recommendations. Our proposed approach deliberately found those kinds of patterns also. In this juncture, identification of noteworthy *sub-global* and *local* recommended items from the forwarded items plays significant role in assessing the importance of items. Also, when an item emerges with higher predicted rating from more than one site, inferring the final recommended rating of those items is also an interesting task. For an example, "item 16" has been emerged as global recommended items from all the three sites, "Site1", "Site2" and "Site3" with the respective rating of 5, 4 and 5. Similarly, "item 18" has been emerged as sub-global recommended items for the two sites, "Site1" and "Site2". If such kinds of patterns are forwarded from participating sites, how to calculate the final recommended rating for those items? To infer the final predicted rating of users for such items, weighting models are to be incorporated to infer final predicted ratings. In such circumstances, factors such as transaction-population of sites, turn-over of branches, cost of the items, location of the branches, and purchasing frequency of users are to be considered as weighting factors for computing the synthesised rating of an item.

5. Conclusion and Scope for Future Work

In this paper, we have proposed a big data recommendation engine framework based on local pattern analytics strategy for mining multi-sourced big data located in different regions. The proposed big data framework efficiently analyse the sources of big data with the advent of commodity hardwares and big data technologies such as Hadoop and Spark. Further it preserves local and global preferences of users by avoiding forceful integration of data sources into a centralised environment. Though the proposed framework segregates the user preferences for local and central level decisions, estimating their preferences with the inclusion of various weighting parameters such as location information of user, trustness of user in rating, assignment of weights for the data sources will further enhance the effectiveness of the proposed framework.

References

Adhikari, A (2013). Clustering local frequency items in multiple databases. *Journal of Information Sciences*, 237, 221–241.

Adhikari, A and PR Rao (2008). Synthesizing heavy association rules from different real data sources. *Pattern Recognition Letters*, 29(1), 59–71.

Adhikari, J, P Rao and A Adhikari (2009). Clustering items in different data sources induced by stability. *International Arab Journal of Information Technology*, 6(4), 394–402.

Adhikari, A, P Ramachandrarao and W Pedrycz (2011). Study of select items in different data sources by grouping. *Knowledge and Information Systems*, 27(1), 23–43.

Ahamed, BB and T Ramkumar (2016). An intelligent web search framework for performing efficient retrieval of data. *Computers & Electrical Engineering*, 56, 289–299.

Bu, J, X Shen, B Xu, C Chen, X He and D Cai (2016). Improving collaborative recommendation via user-item subgroups. *IEEE Transactions on Knowledge and Data Engineering*, 28(9), 2363–2375.

Choi, IY, MG Oh, JK Kim and YU Ryu (2016). Collaborative filtering with facial expressions for online video recommendation. *International Journal of Information Management*, 36(3), 397–402.

Elahi, M, F Ricci and N Rubens (2016). A survey of active learning in collaborative filtering recommender systems. *Computer Science Review*, 20, 29–50.

Geuens, S, K Coussement and KW De Bock (2018). A framework for configuring collaborative filtering-based recommendations derived from purchase data. *European Journal of Operational Research*, 265(1), 208–218.

Gillick, D, A Faria and J DeNero (2006). MapReduce: Distributed computing for machine learning. Berkley, p. 18.

He, D, X Wu and X Zhu (2010). Rule synthesizing from multiple related databases. *In Proceedings of the Fourteenth Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Hyderabad, India, pp. 201–213.

Hu, J, S Sharma, Z Gao and V Chang (2017). Gene-based collaborative filtering using recommender system. *Computers & Electrical Engineering*, 65, 332–341.

Hwangbo, H, YS Kim and KJ Cha (2018). Recommendation system development for fashion retail e-commerce. *Electronic Commerce Research and Applications*, 28, 94–101.

Isinkaye, FO, YO Folajimi and BA Ojokoh (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, 16(3), 261–273.

Luo, D, C Ding and H Huang (2012). Parallelization with multiplicative algorithms for big data mining. *In Proceedings of the 12th IEEE International Conference on Data Mining*, Brussels, pp. 489–498.

Meng, S, W Dou, X Zhang and J Chen (2014). KASR: A keyword-aware service recommendation method on Map Reduce for big data applications. *IEEE Transactions on Parallel and Distributed Systems*, 25(12), 3221–3231.

Nedunchezhian, R and K Anbumani (2006). Post mining — Discovering valid rules from different sized data sources. *International Journal of Information Technology*, 3(1), 47–53.

Palanisamy, V and R Thirunavukarasu (2017). Implications of big data analytics in developing healthcare frameworks — A review. *Journal of King Saud University — Computer and Information Sciences*, Available at https://doi.org/10.1016/j.jksuci.2017.12.007. Accessed on 12 March 2018.

Pereira, R, H Lopes, K Breitman, V Mundim and W Peixoto (2014). Cloud based real-time collaborative filtering for item–item recommendations. *Computers in Industry*, 65(2), 279–290.

Ramkumar, T, S Hariharan and S Selvamuthukumaran (2013). A survey on mining multiple data sources. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 3(1), 1–11.

Ramkumar, T and R Srinivasan (2008). Modified algorithms for synthesizing high-frequency rules from different data sources. *Knowledge and Information Systems*, 17(3), 313–334.

Ramkumar, T and R Srinivasan (2009). The effect of correction factor in synthesizing global rules in a multi-database mining scenario. *Journal of Applied Computer Science & Mathematics*, 3(6), 33–38.

Ramkumar, T and R Srinivasan (2010). Multi-level synthesis of frequent rules from different data sources. *International Journal of Computer Theory and Engineering*, 2(2), 195–204.

Ramkumar, T, R Srinivasan and S Hariharan (2014a). Synthesizing global association rules from different data sources based on desired interestingness metrics. *International Journal of Information Technology & Decision Making*, 13(3), 473–495.

Ramkumar, T, S Hariharan and S Selvamuthukumaran (2014b). Synthesizing global negative association rules in multi-database mining. *International Arab Journal of Information Technology*, 11(6), 526–531.

Ranger, C, R Raghuraman, A Penmetsa, G Bradski and C Kozyrakis (2007). Evaluating mapreduce for multi-core and multiprocessor systems. In *Proceedings of the IEEE 13th International Symposium on High Performance Computer Architecture*, Arizona, pp. 13–24.

Rezaeimehr, F, P Moradi, S Ahmadian, NN Qader and M Jalili (2018). TCARS: Time-and community-aware recommendation system. *Future Generation Computer Systems*, 78, 419–429.

Sarwar, B, G Karypis, J Konstan and J Riedl (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, Hong Kong, pp. 285–295.

Shams, B and S Haratizadeh (2018). Reliable graph-based collaborative ranking. *Information Sciences*, 432, 116–132.

Tsai, CW, CF Lai, HC Chao and AV Vasilakos (2015). Big data analytics: A survey. *Journal of Big Data*, 2(21), 1–32.

Wu, X and S Zhang (2003). Synthesizing high-frequency rules from different data sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(2), 353–367.

Wu, X, X Zhu, GQ Wu and W Ding (2014). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 26(1), 97–107.

Zhang, S, X Wu and C Zhang (2003). Multi-database mining. *IEEE Computational Intelligence Bulletin*, 2(1), 5–13.

Zhang, C, M Liu, W Nie and S Zhang (2004a). Identifying global exceptional patterns in multi-database mining. *IEEE Intelligent Informatics Bulletin*, 3(1), 19–24.

Zhang, S, C Zhang and X Wu (2004b). *Knowledge Discovery in Multiple Databases*. London: Springer Science & Business Media.

Zhang, F, T Gong, VE Lee, G Zhao, C Rong and G Qu (2016). Fast algorithms to evaluate collaborative filtering recommender systems. *Knowledge-Based Systems*, 96, 96–103

Zhang, S, X You, Z Jin and X Wu (2009). Mining globally interesting patterns from multiple databases using kernel estimation. *Expert Systems with Applications*, 36(8), 10863–10869.

Zhang, S and MJ Zaki (2006). Mining multiple data sources: Local pattern analysis. *Data Mining and Knowledge Discovery*, 12(2–3), 121–125.

Zhou, L, S Pan, J Wang and AV Vasilakos (2017). Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350–361.