Pandove, Divya; Malhi, Avleen

A Correlation Based Recommendation System for Large Data Sets

Published in:
Journal of Grid Computing

DOI:
10.1007/s10723-021-09585-9

Published: 01/12/2021

Document Version
Publisher's PDF, also known as Version of record

Journal of Grid Computing (2021) 19:42
https://doi.org/10.1007/s10723-021-09585-9

A Correlation Based Recommendation System for Large Data Sets

Divya Pandove · Avleen Malhi

Received: 24 June 2021 / Accepted: 29 August 2021

Abstract Correlation determination brings out relationships in data that had not been seen before and it is imperative to successfully use the power of correlations for data mining. In this paper, we have used the concepts of correlations to cluster data, and merged it with recommendation algorithms. We have proposed two correlation clustering algorithms (RBACC and LGBACC), that are based on finding Spearman's rank correlation coefficient among data points, and using dimensionality reduction approach (PCA) along with graph theory respectively, to produce high quality hierarchical clusters. Both these algorithms have been tested on real life data (New York yellow cabs dataset taken from http://www.nyc.gov), using distributed and parallel computing (Spark and R). They are found to be scalable and perform better than the existing hierarchical clustering algorithms. These two approaches have been used to replace similarity measures in recommendation algorithms and generate a correlation clustering based recommendation system model. We have combined the power of correlation analysis with that of prediction analysis to propose a better recommendation system. It is found that this model makes better quality recommendations as compared to the random recommendation model. This model has been validated using a real time, large data set (MovieLens dataset, taken from http://grouplens.org/datasets/movielens/latest). The results show that combining correlated points with the predictive power of recommendation algorithms, produce better quality recommendations which are faster to compute. LGBACC has approximately 25% better prediction capability but at the same time takes significantly more prediction time compared to RBACC.

Keywords Correlation clustering · Recommendation system model · RBACC · LGBACC

1 Introduction

Correlation analysis can be defined as an efficacious method to study the relationship among the data points. Strong and weak correlations in data points help in capturing the current and future trends in the data. There are many scenarios where correlations in data have been able to understand seemingly irrelevant data and give accurate results. Over the years, with the help of experimentation and hypothesis, correlation analysis has evolved as a concept. These days, given the power of high powered computing and greater storage capacity, the concept of correlation analysis is applied to large and high dimensional datasets. As a

D. Pandove
Glover Park Group, Washington DC, USA
e-mail: dpandove@gmail.com

A. Malhi (✉)
Aalto University, Helsinki, Finland

A. Malhi
Bournemouth University, Bournemouth, UK
e-mail: amalhi@bournemouth.ac.uk

Published online: 18 October 2021

result there is rapid emergence of correlations in data, with not much cost. The correlation methods aim on finding correlations with focus on "what" and not the "why" aspect of the data. The "why" part of data might be very appealing and interesting to the human mind, but does not help in generating useful insights about data points' relationships. The main idea is to find the correlations and patterns between data points rather than studying the cause-effect relationships which will help to visualize the links among data, unseen before. The premise of this approach is that causality can be rarely proven [1]. Correlations not only help to analyze the small data sets but they can be fairly used with high dimensional data. Now-a-days, important tools are being developed by experts for identifying and comparing non-linear correlations. The techniques of analysis are being aided and improved by demanding novel methods, and softwares to extract non-causal relationships among data [2]. Given increasing use of intensive data collection techniques these days, there is a substantial increase in the number of data points, as well as number of features, in the datasets. Feature extraction techniques applied on these datasets may not be very effective, as the features extracted may either have false correlations among each other or may be noisy. These irrelevant features must be disregarded by applying appropriate data mining techniques for the selection of relevant features. If a set of 'n' documents are given to cluster them into topics and one has no information about what the topic is. If we have a classifier $f(A, B)$ such that if two documents A and B are given, it outputs its belief if A and B are similar to each other or not where, the behavior of f is being learned from past training data. Therefore, the most intuitive technique for clustering in this case is that function f is applied to each pair of documents to find a clustering which agrees with results in a maximum possible way. The most prominent challenge while clustering high-dimensional data space is that the feature relevance is dependent on the clusters they belong to. Moreover, there might be relevance of correlations among dataset's different attributes with disparate clusters. This fact of relevance of feature correlation with various clusters is termed as local feature selection [3]. In a given dataset, the correlation detection between different features is a primary task of data mining. The higher degree of collinearity in a dataset means high correlation of features, corresponding to the fact that there exist approximate linear dependencies among two or more attributes. Complex dependencies may exist such that there might be dependence of one or more features on a combination of various features. If correlations are known initially, the dimensionality of dataset may be reduced by elimination of redundant features. A new concept of knowledge discovery in databases has been introduced which is known as correlation clustering [4] for the detection of dependencies that exist among features, and to cluster the points having common pattern dependencies. Correlation clusters are formed by grouping the data into subsets with the condition that the data points in the same correlation cluster are linked with common hyperplane of arbitrary dimensionality. There is also requirement for correlation cluster algorithms to present certain density known as feature similarity. The concept of correlation clustering has been used in various application domains [5–7].

1.1 Contributions

The contributions of the paper are four-fold:

1) we have introduced two algorithms for obtaining high quality correlation clusters. These algorithms are similar in the sense that they produce high quality clusters but they are different in the concepts they are based on.
2) The first algorithm is based on finding correlation coefficients among given data points, and clustering data based on these values.
3) The second algorithm is based on merging the technique of data reduction with the concepts of graph theory and hierarchical correlation clustering. This approach is suitable for finding correlation clusters in high dimensional data.
4) We have combined the power of correlation analysis with that of prediction analysis to propose a better recommendation system.

These two approaches have been used to replace similarity measures in recommendation algorithms and generate a correlation clustering based recommendation system model. It is found that this model makes better quality recommendations as compared to the random recommendation model.

1.2 Organization

The rest of the paper is organized as: Section 2 gives a brief summary of the work done in the related area of correlation clustering. Section 3 mathematically defines and interprets correlation clusters. Section 4 enlists the preliminaries required for understanding the proposed algorithms. Section 5 introduces and explains the proposed correlation clustering algorithms and experimental evaluations for validating the proposed approaches. Section 6 discusses the framework of the recommendation system model. It also contains the preliminaries for the proposed framework, followed by a detailed explanation and experimental evaluation of the model. Section 7 briefly concludes the work presented in this paper.

2 Related Work

Correlation clustering is a type of subspace clustering. It selects the number of clusters automatically to give approximated solutions. This method is used to handle instances with the focus on relationships among the objects, rather than actual object representations. Correlation clustering has marked as a paramount inclusion data mining field considering the ever increasing data scale these days. The general idea is to discover a clustering that either minimizes disagreements or maximizes agreements between data points. Many methodologies, like linear programming formulations, approximations etc., are being used as an approach for this problem. The problem of correlation clustering can be defined as to find a partition of vertices in the form of clusters to agree with labels of edges as much as possible in a fully connected graph with labelled edges as '+' for similarity and '-' for difference [4]. The main aim is to maximize the positive edges in the desired partition. The positive edges sum in each cluster and negative edge sum between clusters need to be maximized. This method of clustering does not take the cluster count as clustering parameter. Edge labels define the optimal cluster number between 1 and n. We categorize the correlation clustering algorithms by the dimensionality reduction approaches they are based on.

2.1 PCA Based Approaches

It covers the major section of correlation clustering approaches. A local correlation dimensionality based hierarchical approach, HICO [8] was proposed for defining the distance between data points and for calculating data's subspace orientation. Another approach, ORCLUS [6] assigns data objects to first k seeds based on the eigenvector distance function.The efficiency of the algorithm is increased by choosing the highest value of k. A variant of ORCLUS was given by Li et al. [9] which is more efficient as it can produce correlation clusters of high quality from noisy data. In 4C algorithm [5], the cluster expands around the seed until density criteria is fulfilled which is specified by the upper bound on the data points that can lie within defined neighborhood (distance matrix from 2 data point's Eigen systems) of points and there is no decision on cluster number in advance. COPAC [10] is an improvement of 4C algorithm by reducing the time complexity to d^2. It works by partitioning the data space such that search is limited to local correlation dimensionality clusters. Eric [11] is another approach which takes affine distance into consideration where approximate linear dependency of each data point defines the neighborhood. As a result, subspace cluster hierarchy is formed.

2.2 Hough Transform Based Approaches

In Hough transform, trigonometric function is used to represent each data point's link with infinite points. A global subspace clustering approach is obtained by hough transform [12]. In CASH algorithm [13], dense regions are carved out by using grid based methodology. Attributes are used to divide the data space and calculating the functions which intersect the hyperboxes created due to data space division.

2.3 Other Approaches

There is another approach called CURLER [20] where the non-linear and arbitrary correlations are detected. Micro clusters are used in this approach which uses an EM variant for generation and are finally grouped to find correlation clusters. There need not exist linear

relationship among the clusters. Moreover, there is an assumption that every data object belongs to each cluster but is assigned different probability to each cluster. In a short span, many subspace clustering algorithms [5, 14–17] have been adduced for finding out clusters in data space's axis-parallel projections. The local data is not captured in these algorithms to find correlated objects' clusters as there is arbitrary orientation of correlated data's principal axes. Finding subgroups having similar trend in attributes' subsets is done in pattern-based clustering methods [7, 18–20] which is also called by bi-clustering or co-clustering. There exists unique form of clustering in pattern based clustering having all positively correlated attributes and excluding the negative correlations as well as correlations having dependency on two or more than two attributes.

There are not many techniques in the literature that successfully use the power of correlations for data mining. we propose two correlation clustering algorithms based on entirely different techniques. The first one is based on calculating mathematical correlations between data points, while the other is based on the definition of correlation clusters. Hence, we propose a recommender system model based on the two proposed algorithms which merges the power of correlation clustering with that of the prediction analysis to make better quality recommendations.

3 Understanding Correlation Clusters

This section describes the correlation clusters formalization and the things which need to be considered in their interpretation.

3.1 Eigenvectors: Strong and Weak

Let $\mathcal{V}$ be the Eigenvectors and they are partitioned into two classes: strong and weak (denoted by $\mathcal{S}$ and $\mathcal{W}$. They satisfy following conditions

1) $\mathcal{S}, \mathcal{W} \subseteq \mathcal{V}$
2) $\mathcal{S} \cup \mathcal{W} = \mathcal{V}$
3) $\mathcal{S} \cap \mathcal{W} = \phi$

In non-technical terms, those Eigenvectors that correspond to the least variance in the corresponding dimension (with Eigenvalue $\lambda \approx 0$) are termed as weak Eigenvectors. Without loss of generality, let us assume that first s vectors are strong and rest of them are weak. Now, either the cluster hyperplane can be defined by strong Eigenvectors, or they can also be defined by the weak Eigenvectors, perpendicular to this hyperplane. There is an additional benefit of working with weak Eigenvectors. It allows to examine the numerical dependencies between the attributes.

3.2 Defining Correlation Clusters

Suppose that there exits an s-dimensional cluster C in the original data space D of d dimensions. Clearly $C \subseteq D$ and $\{s, d - s\}$ be the strong and weak Eigenvectors for the cluster C. Thus s-dimensional hyperplane for the definition of cluster C is possible through mean $\mu = (\overline{x}_1, \ldots, \overline{x}_s)^T$. Now, they also correspond to the weak eigenvectors $\mathcal{W}$ which are orthogonal to the hyperplane. Thus the hyperplane can be defined using the following equation system:

$$V \Delta X = 0 \tag{1}$$

Here,

$$V = \begin{bmatrix} v_{s+1,1} & \ldots & v_{s+1,d} \\ v_{d,1} & \ldots & v_{d,d} \end{bmatrix}, \Delta X = \begin{bmatrix} x_1 - \overline{x}_1 \\ \ldots \\ x_d - \overline{x}_d \end{bmatrix},$$

$$0 = \begin{bmatrix} 0 \\ \ldots \\ 0 \end{bmatrix}$$

To find out dependencies in weak Eigenvectors, Gauss Jordan elimination method [21] is used with total pivoting, which is numerically stable and simple to implement. Although the cluster model works for all data points $x \in C$ but it can also work as predictive testing for the new incoming data. Now for every cluster $C_i \subseteq D$ the following calculations are done:

1) Calculate covariance matrix $\sum_i$
2) Choose weak Eigenvectors W_i from the covariance matrix with reference to a given threshold value τ
3) Solve the hyperplane equation $W^T.x_i = W^T.\overline{x}_i$ for cluster C_i
4) Perform Gauss Jordan elimination and row reduction echelon form, to solve for mutual dependencies quantitatively

3.3 Interpretation of Correlation Clusters

Let the solution obtained from above system in five dimensional space be the following:

$$\begin{bmatrix} 1 & 0 & \alpha_3 & 0 & \alpha_5 \\ 0 & 1 & \beta_2 & 0 & \beta_5 \\ 0 & 0 & 0 & 1 & \gamma_5 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix}$$

We find out the strong eigenvectors which are actually free attributes with the linear dependence among the following vectors for the given values of constants $\alpha_2, \alpha_5, \beta_2, \beta_5, \gamma_5$: (a) $w_1 \sim w_3 \sim w_5$ (b) $w_2 \sim w_3 \sim w_5$ (c) $w_4 \sim w_5$. Now, to understand this relationships, often the domain knowledge of the experts are a prerequisite. The dependence of Eigenvector can also be highlighted through experimentation. For example increasing or decreasing values of $\{w_1, s_3, w_5\}$ that is choosing one vector from each equation and watching the effect in other variables will alarm some patterns. Now, further refinement could be done by simply trimming the related related models generated by dependent Eigenvectors. A set of independent vectors may also be represented by a novel fourth variable. Hence, modeling the sections of a big and complex structured systems and performing the experiments to examine their outcomes may be extremely useful for analyzing the nature of causal relationships among Eigenvectors.

4 Preliminaries

In this section, we present the concepts and attributes to describe the proposed algorithms.

- PCA (Principal Component Analysis): PCA is a dimensionality reduction technique that aims at constructing data vectors that indicate data variance in the given data. Once the data is imported into the computing environment, it is standardized and the covariance matrix is computed. Subsequently, Eigenvalues and Eigenvectors are calculated, based on the covariance matrix. These values are then arranged in decreasing order, and the components lower down the list are ignored in order to obtain the principal components. These components are then arranged to form a feature vector matrix. This matrix is then used to obtain final data by using the formula: *Final data = Row Feature Vector X Row Data Adjust*. Here, Row Feature Vector is the matrix with the Eigenvectors in the columns transposed so that the Eigenvectors are in the rows, with the most significant Eigenvector at the top, and is the mean-adjusted data transposed, i.e. the data items are in each column, with each row holding a separate dimension. The final data set is derived by taking data items in columns and dimensions along the rows.
- Local covariance matrix: It ($\sum_P$) is the covariance matrix of the k nearest neighbors of point P. Given that $k \in N$, $k \leq |D|$. The local covariance matrix $\sum_P$ of a point $P \in D$ w.r.t k is formed by r nearest neighbors of P. Let $\overline{Y}$ be the centroid of $NN_r(P)$, then [22]:

$$\sum_P = \frac{1}{|NN_r(P)|} \cdot \sum_{x \in NN_R(P)} (Y - \overline{Y}).(Y - \overline{Y})^T$$

- Correlation similarity matrix of a point: Let us consider a point P, such that $P \in D$. V_P and E_P are the the Eigen vectors and Eigen values of the point P respectively. Let C be a constant with $C \geq 1$. Now in order to calculate a new Eigen value matrix with $\hat{E_P}$ having diagonal entries $\hat{e_i}$.

$$\hat{e_i} \begin{cases} 1 & if \quad \Omega(e_i) > \delta \\ C & if \quad \Omega(e_i) \leq \delta \end{cases} \tag{2}$$

Here, Ω represents normalization of the Eigenvalues to convert into [0, 1]. The correlation similarity matrix is represented as: $\hat{B_P} = V_P.\hat{E_P}.V_P^T$. The correlation similarity matrix measure associated with point P is denoted by:

$$cdist_P(P, O) = \sqrt{(P - O)^T.\hat{B_P}(P - O)}$$

- Affinity matrix: t is made up of positive values, and is a type of a symmetric matrix. It represents affinity scores between the two objects. It is computed by applying k-nearest neighbor in order to build a matrix of closest data points. Affinity matrix replaces clustering by a graph partition problem. Here, graph components are interpreted as clusters. The graph construction must be partitioned in a way that the edges connecting different clusters should have low weights. Even the edges within the same clusters must also have high values [23]. We consider 3 nearest neighbors for each point, including the point as well. Given the similarity matrix as an input, the output of the

clustering contains at most three vertices of the graph G'. Here, G' is a complete graph formed out of a larger graph H.

Given 2 data points x_i and x_j, Affinity $A_{i,j}$, that is positive and symmetric, and depends on the Euclidean distance $\|x_i - x_j\|$ between the two data points. Affinity matrix is defined as:

$$A_{ij} \simeq exp(-\alpha \|x_i - x_j\|^2) \quad (3)$$

Where, α is a constant.

- Graph Laplacian transform: The graph Laplacian of graph G, L_G has Eigenvalues $\{\lambda_1, \lambda_2, ..., \lambda_n\}$ where, $\lambda \leq \lambda 2 \leq ..... \leq \lambda_n$ and Eigenvectors $\{v_1, v_2, ..., v_n\}$. Eigenvalues reveal global graph properties which is not apparent from edge structure. If 0 is the Eigenvector of L with K different Eigenvectors, i.e., $0 = \lambda_1 = \lambda_2, ..., = \lambda_k$, then G has K connected components. If the graph is connected, $\lambda \geq 0$ and λ_2 is the algebraic connectivity of G. The greater the value of λ_2 , the more connected G is [24]. An unnormalized Graph Laplacian given by $U = D - A$ and a Simple Laplacian is given by $H = D^{-1}A$, where A is the affinity matrix and $D^{-1}A$ is a transition matrix.
- Hierarchical Agglomerative clustering: Hierarchical Agglomerative clustering based on correlation coefficient is an application of correlation analysis. The result of a hierarchical algorithm is usually described in the form of a dendrogram, where data set is represented by its root and data object by each leaf node. AGNES (Agglomerative Nesting) [25] is the hierarchical clustering approach used in this paper. In this approach, the user does not specify a value of k (number of clusters). This algorithm constructs a tree like hierarchy which implicitly contains all values of k. When one has to perform cluster analysis on a set of observed variables, then one can use parametric or non parametric correlation coefficient of two variables and convert them into a dissimilarity matrix. The data points are fussed together based on this matrix to form data clusters following an Agglomerative approach. The authors have used a graphical representation known as banner to represent clustering by AGNES. It looks like a waving flag and consists stars and stripes. The stars indicate linking of objects and stripes are repetitions of labels of these objects. A banner helps to easily navigate the structure of data set at a given level. This means at a particular value of k, a banner shows how data points are clustered and placed. The steps of Agglomerative hierarchical clustering is given in Algorithm 1.

Algorithm 1 Agglomerative hierarchical clustering.

Require: A Set of data points $X = \{x_1, x_2, ..., x_n\}$

1: Begin with disjoint clustering defined by:
   a) Level $L(0) = 0$
   b) Sequence number m=0
2: Find least distance pair of clusters in pair r and s as:
   $d[(r), (s)] = mind[(i), (j)]$
3: **while** $L(m) \neq d[(r), (s)]$ **do**
4: Merge clusters (r) and (s)
5: $m = m + 1$
6: **end while**
7: Update distance matrix D by:
   a) Delete rows and columns corresponding to clusters (r) and (s)
   b) Add a row and column corresponding to newly formed cluster
   c) Distance between new cluster (r,s) and old cluster (k) is given by:
   $d[(k), (r, s) = min(d[(k), (r)], d[(k), (s)]$
8: **if** all points in one cluster: **then**
9: Stop
10: **else**
11: Go to Step 2
12: **end if**
13: Exit

- Agglomerative coefficient: It is calculated from the banner produced as a graphical output of the algorithm AGNES. For each object i, the line containing its label is identified and its length $l(i)$ is measured on the 0-1 scale above and below the banner. The coefficient of the data set is defined as [25]:

$$AC = \frac{1}{n} \sum_{i=1}^{n} l(i) \quad (4)$$

The AC is a dimensionless quantity between 0 and 1 as $l(i)$ lies between 0 and 1. It is the average width of the banner and it remains the same even when all the original dissimilarities are multiplied by a constant factor. AC defines the strength of

the clustering structure that has been obtained by group average linkage. If $d(i)$ denotes the dissimilarity of object i to the first cluster it is merged with divided by the dissimilarity of the merger in the last step of the algorithm. Then Agglomerative coefficient is given by average of all $1 - d(i)$.

- Fowlkes-Mallow index: It is an external evaluation method to determine the quality of a cluster. It is a measure of similarity, and a comparison is made between the given clustering and hierarchical clustering or the given clustering and a benchmark classification. A higher value of the FM index indicates a greater similarity between the clusters and the benchmark classifications. FM index can be calculated as follows [26]: We consider two hierarchical clusterings for n objects which can be labeled as A_1 and A_2. The trees A_1 and A_2 which are obtained as a result of this clustering can be cut to produce $k = 2, ..., n - 1$ clusters for each tree. For each value of k we obtain corresponding values of m given by:

$$M = [m_{i,j}] \tag{5}$$

Here $i = 1, ..., k$ and $j = 1, ..., k$, $m_{i,j}$ gives the common objects between i^{th} cluster of A_1 and j^{th} cluster of A_2. The Fowlkes-Mallows index for a given value of k can be defined as:

$$B_k = \frac{T_k}{\sqrt{P_k Q_k}} \tag{6}$$

Here, $T_k = \sum_{i=1}^{k} \sum_{j=1}^{k} m_{i,j}^2 - n$
$P_k = \sum_{i=1}^{k} (\sum_{j=1}^{k} m_{i,j})^2 - n$
$Q_k = \sum_{j=1}^{k} (\sum_{i=1}^{k} m_{i,j})^2 - n$

- Spearman's rank correlation coefficient [27] is a rank based correlation coefficient, given by:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{7}$$

where, $d_i = x_i - y_i$ is the difference between ranks of observations. The value of the correlation coefficient lies between -1 and 1 and assesses monotonic relationships between the rank values of those two variables.

- Correlation and dissimilarity matrix: Once the data is imported in the computing environment, ρ is used to compute correlation coefficients between paired values and generate a correlation matrix. The values in this matrix lie between -1 and 1. This correlation matrix acts as an input for the distance or dissimilarity matrix. Dissimilarities are always positive and are represented as $d(i, j)$. They are small when i and j are close to each other and become large when i and j are very different. It is assumed that dissimilarities are always symmetric and dissimilarity of an object to itself is zero. In this paper, dissimilarities are calculated from the correlation matrix using the formula [25]:

$$d(f, g) = 1 - \mid R(f, g) \mid \tag{8}$$

where, R(f,g) is the correlation coefficient value.

5 Proposed Correlation Clustering Algorithms

We propose two correlation clustering algorithms based on entirely different techniques. The first one is based on calculating mathematical correlations between data points, while the other is based on the definition of correlation clusters, as defined in the previous sections.

5.1 Rho Based Correlation Clustering (RBACC)

We present a correlation clustering based algorithm that does not require to specify the number of clusters in advance. The algorithm creates a tree like hierarchy of data points. This is a agglomerative hierarchical clustering algorithm based on mathematical correlation between data points. The value of correlation is given by Spearman's rank correlation coefficient (ρ). The focus of this approach is to produce high quality clusters. The detailed steps for RBACC are given in Algorithm 2. The first step of the algorithm converts data in the form of a data matrix. This is done to give a structure to the data, as the inputed data may also be semi-structured or unstructured. This data matrix is then used to generate a correlation matrix using 'ρ.' Pairs of attributes are selected from the matrix and correlations are found between them. A dissimilarity matrix is generated from this matrix, that acts as the input for the agglomerative hierarchical clustering algorithm (AGNES). The next step is to check the quality of the clusters produced. We use Agglomerative Coefficient and Fowlkes-Mallow index for this purpose. This algorithm is a simple algorithm that has

evolved around the idea that the clusters generated by determining the correlations between data points are high quality and distinctive in nature.

Algorithm 2 ρ based correlation clustering algorithm (RBACC).

Require: Data Matrices A[i][i], B[i][i]
length of A,B = n
Assumption: Input data is represented as a data matrix
Ensure: High Quality Clusters
1: $Rank_i$ is calculated as: R_i = A[i][1], S_i = B[i][1]
2: Correlation coefficient is calculated as:
ρ_i = 1-6 $\times(\frac{\sum_{i=1}^{n}(R_i - S_i)^2}{n(n^2-1)}$
3: Calculate t-statistic:
4: **if** (ρ==1) or (ρ==-1) **then**
5: $t = r * \infty$
6: **else**
7: $t = \rho \times \sqrt{\frac{N-2}{1-\rho^2}}$
8: **end if**
9: Calculate P value:
P = 2$\times$ $(1 - cdf^*(abs(t), N - 2))$
10: Generate correlation matrix C[4][n] with dimensions: A[i][1], B[i][1], ρ_i, P_i
11: Calculate dissimilarity matrix using D = $1 - \rho$
12: Generate clusters using AGNES Refer Algorithm 1
13: Check cluster quality using:
a) Agglomerative coefficient (AG) = $\frac{1}{n}\sum_{i=1}^{n} l(i)$
b) Fowlkes-Mallow index $B_k = \frac{T_k}{\sqrt{P_k Q_k}}$
c) Visulaization plots
cdf: Cumulative distributive function

5.2 Locality Assumption Graph Based Correlation Clustering (LGBACC)

In high-dimensional data, clusters often exist in the form complex hierarchical relationships. In order to explore these relationships, there is a need to integrate dimensionality reduction techniques with data mining approaches and the graph theory. We have proposed an algorithm that integrates the basic elements of PCA with those of graph theory to produce high quality hierarchical clusters. This algorithm is based on locality assumption. This means that it is assumed that all the clusters are present on a common hyperplane. The detailed steps of this approach are given in Algorithm 3. Once the data is imported into the computing environment, it is standardized. This standardized data is used to compute the local covariance matrix. These data pre-processing steps help to get data in a standard structured format. The next step is to reduce the dimensionality of the covariance matrix. We use PCA to perform this operation. A new data set $S2$ is derived after selected principle components from the Eigenvector and Eigenvalue matrices, calculated from the covariance matrix. To integrate the reduced data with graph theory, we compute local covariance matrix. From this matrix, similarity and subsequently affinity matrices are generated. In the next step, simple graph laplacian is applied and a new dataset $S2$ is generated. This dataset is used to generate clusters using AGNES. The quality of these clusters is checked using Agglomerative Coefficient and Fowlkes-Mallow index.

Both the proposed approaches are similar in the manner that they produce quality clusters, and overcome the problem of specifying the number of clusters in advance. There are many points of differentiation as well. The comparison of RBACC and LGBACC is given in Table 1 and points of differentiation between different correlation clustering approaches are given in Table 2.

5.3 Experimental Evaluation

To test the working of the proposed algorithms on the basis of the stated parameters, we have performed experiments on four datasets, using SparkR (R on Spark). Spark R is an R language package that provides a front end to use Apache Spark from R. It provides a distributed data frame implementation that supports different operations but on large datasets. It also supports distributed machine learning using MLib. We have used Amazon simple storage (S3) buckets to store our large datasets and piped the data into the SparkR environment using the datasource API. The SparkR architecture used for experimentation is given in Fig. 1. We have used the New York yellow cabs dataset taken from http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. This is a very large dataset with approximately 1.3 billion data points and takes approximately 260 GB on the disk. The data has been taken from January 2009 to December 2016. We have worked

Table 1 Comparison between the two proposed algorithms

	LGBACC	RBACC
Defining features	1. Works on standardized data	1. Based on calculating mathematical correlations between data points.
	2. Uses PCA to reduce dimensions	2. Uses agglomerative clustering to produce clusters
	3. Uses graph theory to obtain better cluster quality	3. Tests the quality of clusters produced
Suitable for data types	Qualitative and quantitative data	Numerical data, both discrete and continuous. Works best on ordinal, interval or ratio variables
Advantages	1. It is more general in nature as it works on similarity matrix data.	1. Does not assume number of clusters in advance
	2. Very effective in handling high dimensional data	2. Produces high quality clusters
	3. Dimensionality reduction reduces the execution time.	3. An ordering of the objects is obtained
	4. It works well on overlapping clusters	4. Mathematical determination results in distinct clusters
Limitations	1. Multiple mathematical operations increase the computational cost	1. Suitable only for numerical data
	2. Dimensionality reduction may result in loss of veracity of data	2. Works only on data matrix
	3. It is based on locality assumption	3. Like all hierarchical clustering algorithms has high time complexity
		4. Mathematical operations increase the time and resource utilization cost.

on 21 attributes combined from all the tables in the database. These attributes are: medallion number, license, vender_id, rate_code, store_and_fwd, pickup_datetime, dropoff_datetime,passenger_count, trip_time_in_secs, trip_distance, pickup_longitude, pickup_latitude, dropoff_longitude, dropoff_latitude, payment_type, fare_amount, surcharge, mta_tax, tip_amount, tolls_amount and total_amount. We performed experiments on this large dataset to compare the two algorithms on the basis of Agglomerative Coefficient and Fowlkes-Mallow index. The results are given in Table 3. We discover that LGBACC generates better quality clusters than RBACC. Though both of them produce pretty high quality clusters, LGBACC is a little better than RBACC.

The visualization of experiments performed on LGBAAC and RBACC are given in Figs. 2 to 7. All these figures represent a data sample of 23 weeks of taxi activity, chosen randomly from the dataset. This has been done for clear visibility of the data points. Figure 2 shows a correlation plot depicting correlations between clusters formed on the basis of taxi activity on each day of the week. It can be inferred that neither a positive not a negative correlation exists between these clusters. Figure 3 shows data clustering by RBACC, it is represented in the form of bar plots. Figure 4 is a more systematic representation of this clustering in the form of a line graph. The clusters are clearly represented with the help of different colored lines and the variation in data points is shown by the slop of the line. Figure 5 represents clustering performed on the first half of the LGBACC algorithm, before performing PCA. Figure 6 shows percentage of explained variance against the dimensions, to perform PCA on the dataset. Figure 7 is representation of the variance factor map for PCA. The Figures show steps of PCA in detail. These representations are of the PCA phase of the LGBACC algorithm. Figure 8 shows 3D representation of the hierarchical clustering performed on the dataset produced after performing PCA (S2). Figure 9 shows a cluster plot of the distinct quality clusters produced by LGBACC algorithm.

Table 2 Comparison of the proposed clustering algorithms with state-of-the-art algorithms

	Complex Correlation	Simple positive correlation	Simple negative correlation	Dimesionality reduction	Hierarchical structure	Independent order of attributes w.r.t	Independent order of objects w.r.t	Noise robust
ORCLUS	y	y	y	y	–	y	–	–
4C	y	y	y	y	–	y	y	y
COPAC	y	y	y	y	–	y	y	y
ERIC	y	y	y	y	y	y	y	y
CASH	y	y	y	–	y	–	y	y
LGBACC (proposed)	y	y	y	y	y	y	y	y
RBACC (proposed)	y	y	y	–	y	y	y	y

Algorithm 3 Locality assumption graph based correlation clustering algorithm (LGBACC).

Require: Data Matrices A[i][i], B[i][i]
length of A,B = n
Assumption: Input data is represented as a data matrix
Ensure: High Quality Clusters
1: Standardize data (mean=0, variance=1) a) $\overline{a}$ = Mean (A[i][1]) =0, $\overline{b}$ = Mean (B[i][1]) =0
b) σ^2 (A[i][1]) = σ^2 (B[i][1]) = 1
2: Compute local covariance matrix using:
Cov (A[i][1], B[i][1])= $\frac{\sum_{i=1}^{n}(A[i][1]-\overline{a})(B[i][1]-\overline{b})}{n-1}$
3: Apply PCA on local covariance matrix
a) Calculate Eigenvectors and Eigenvalues using:
$B = [b_{ij}], with\ b_{ij} = \sum_{a\in A}(a_i - \overline{a_i}).(a_j - \overline{a_j})$
b) Select principal components
c) Derive new data set S1
d) Compute local covariance matrix using:
Cov (A[i][1], B[i][1])= $\frac{\sum_{i=1}^{n}(A[i][1]-\overline{a})(B[i][1]-\overline{b})}{n-1}$
4: Perform graph theory analysis as:
a) Generate local similarity matrix $S_{i,j}$ = s(a_i, b_i) from S1,
represented as: $\hat{B_P} = V_P.\hat{E_P}.V_P^T$.
b) Generate affinity matrix using:
$A_{ij} \simeq exp(-\alpha\|x_i - x_j\|^2)$
c) Apply Simple Graph Laplacian as:
$L = D - A$
d) Calculate Eigenvector matrix
e) Derive new dataset S2
5: Generate clusters using AGNES Refer Algorithm 1
6: Check cluster quality using:
a) Agglomerative coefficient (AG) = $\frac{1}{n}\sum_{i=1}^{n} l(i)$
b) Fowlkes-Mallow index $B_k = \frac{T_k}{\sqrt{P_k Q_k}}$
7: Exit

We have used the look-alike model to extend this dataset and generate four different datasets, to test the scalability of the proposed approaches, and compare them to the existing hierarchical clustering algorithms. The details of the extended datasets in terms of data points and size are given in Table 4. We have used

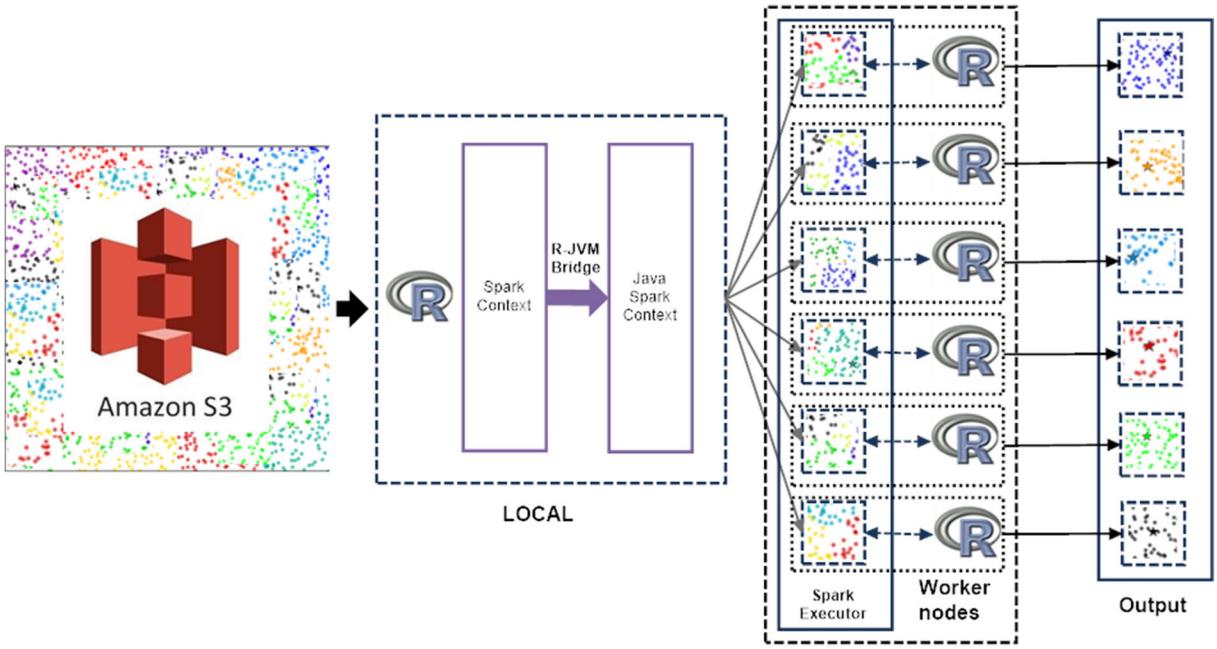

Fig. 1 SparkR architecture used for implementation

the execution time parameter to run this analysis. Different number of worker nodes have been used to compare the algorithms, varying between 5 and 25. The results of this analysis are given in Fig. 10. A linear relationship is observed between the execution times of the algorithms. It is observed that LGBAAC has the fastest execution time as compared to other algorithms. For DS1, at 25 nodes, it is 19% faster than RBACC and 35.3% faster than the single link hierarchical clustering algorithm, which takes the maximum time out all the algorithms considered for this experiment. For DS3, at minimum number of nodes, 5, it is 4.2% faster than RBACC algorithm, but is 5.3% faster, at 20 nodes, of the same dataset. For the largest dataset (DS4), LGBACC is again the fastest at all the nodes and average hierarchical clustering algorithm is a close second with RBACC trailing very closely behind. It was observed that LGBACC and average link hierarchical clustering are the fastest out of the five algorithms. LGBACC, is a little faster than the average link, this is due to the time reduction achieved by applying PCA. RBACC is better than Single and Complete link hierarchical clustering algorithms, in terms of execution time.

6 Recommendation System Model

We now propose a recommender system model based on the proposed algorithms. This model merges the power of correlation clustering, with that of prediction analysis to make better quality recommendations. The flow of the model has been given in Fig. 11.

Table 3 Comparison of cluster quality for the proposed algorithms showing that LGBACC generates better quality clusters

Algorithms	Agglomerative coefficient	Fowlkes Mallows Index		
		Index	E_FM	V_FM
LGBACC	0.9944	1	0.2340068	0.00047908
Rho based CC	0.7376	0.831	0.000465811	0.2457143

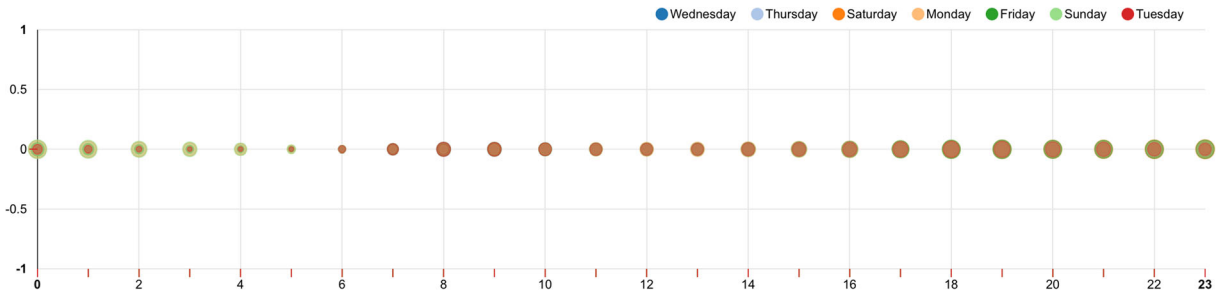

Fig. 2 Correlation plot showing correlations among the clusters formed on the basis of each day's taxi activity

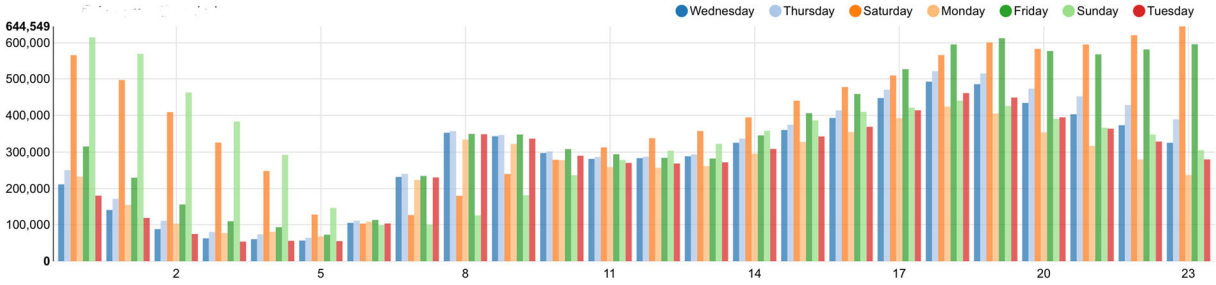

Fig. 3 Clustering by RBACC in the form of bar graphs

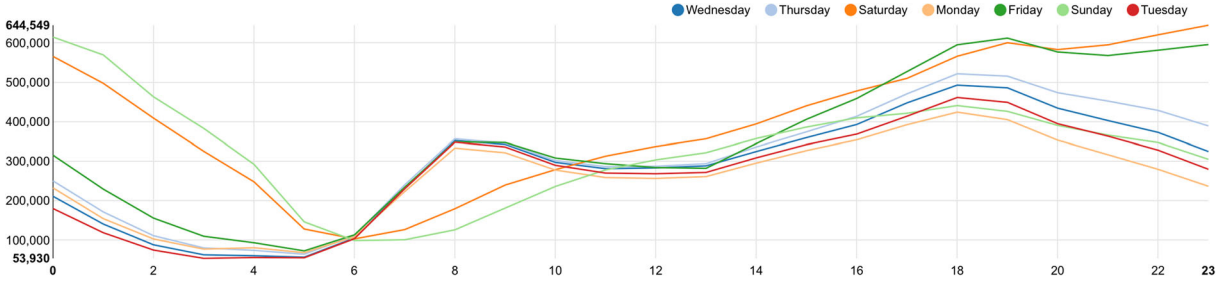

Fig. 4 Clustering by RBACC in the form of line graphs

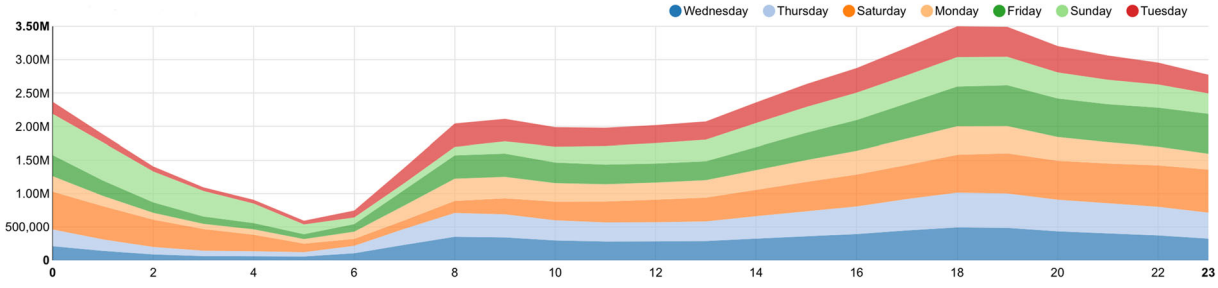

Fig. 5 Clustering on first half of LGBACC, before performing PCA

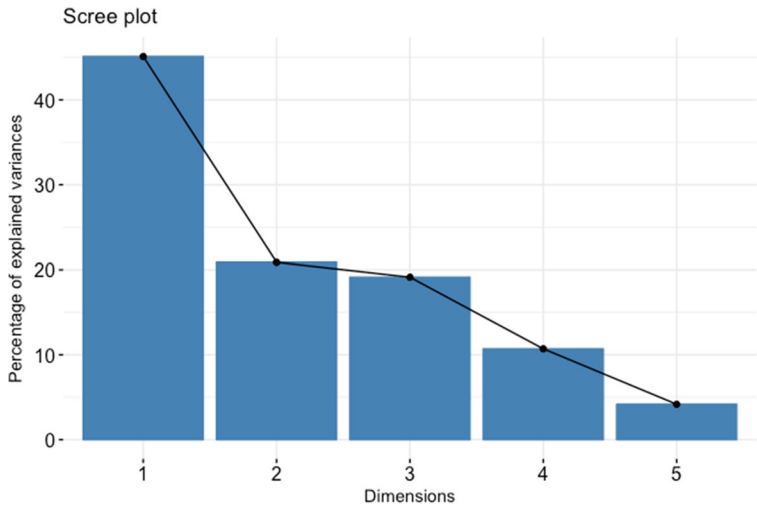

Fig. 6 Percentage of variance v/s Dimensions before PCA

The measures and terms used in the model are given below:

6.1 Preliminaires

- Recommendation algorithms: There are many algorithms that can be used to make recommendations.

We list the algorithms defined in the recommenderlab package of R language:

– User based collaborative filtering (UBCF) [28]: It analyzes rating data collected from many individuals. The assumption is that the users with similar preferences

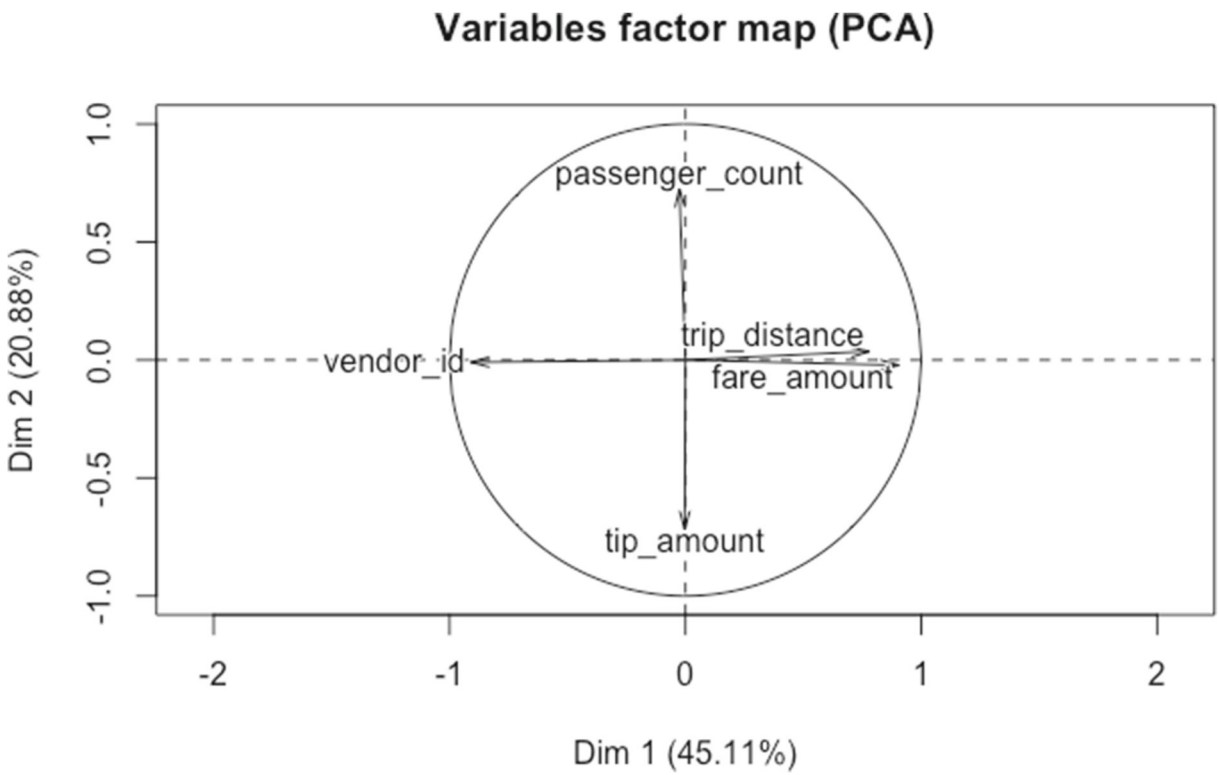

Fig. 7 Variance factor map for PCA on LGBACC algorithm

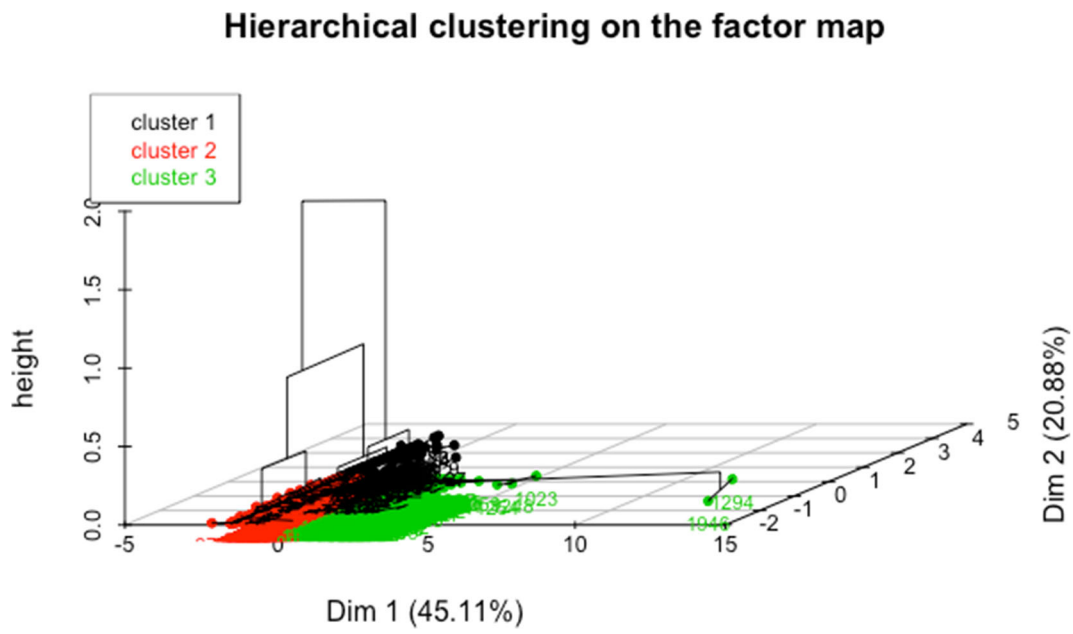

Fig. 8 3D representation of the hierarchical clustering on the factor map after PCA

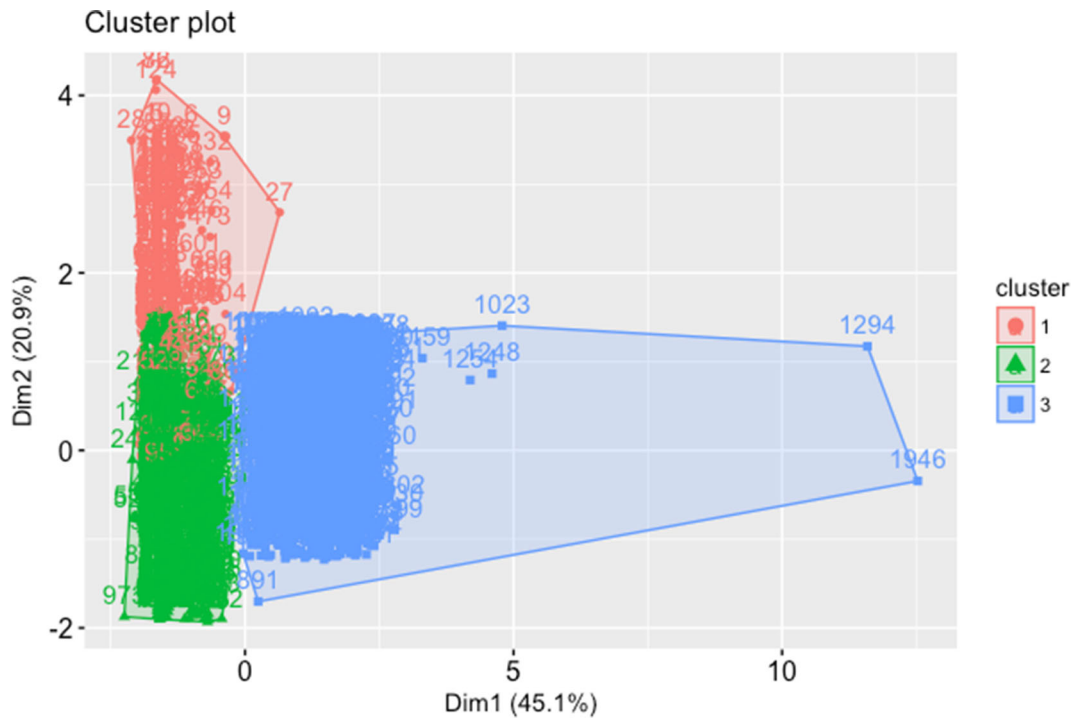

Fig. 9 Cluster plot of the distinct clusters produced by LGBACC

Table 4 Description of datasets used for evaluation

Dataset	Size (approx.)	No. of data points (approx.)
DS1	260 GB	1.3 billion
DS2	520 GB	2.6 billion
DS3	780 GB	3.9 billion
DS4	1.04 TB	5.2 billion

rate items similarly. The missing ratings can be predicted by finding a cluster of similar users and aggregating the ratings to make predictions. The clusters are defined by using similarity measures that either give maximum points of similarity between the users or takes all the users above a particular threshold.

- Item-based collaborative filtering (IBCF) [29]: This approach is based on a rating matrix. The recommendations are made based on items that can be inferred from the ratings matrix. This approach is based on the assumption that users will go for items similar to the other items they have liked.
- User and Item-based collaborative filtering using 0-1 data [30]: This method is used in situations where less rating data is available. The usage behavior is analyzed to infer preferences. The information is presented in the form of 0's and 1's. 1 means that user has a preference for a product and 0 means not.
- Recommendations for 0-1 data based on association rules [31]: The recommendations are made based on the

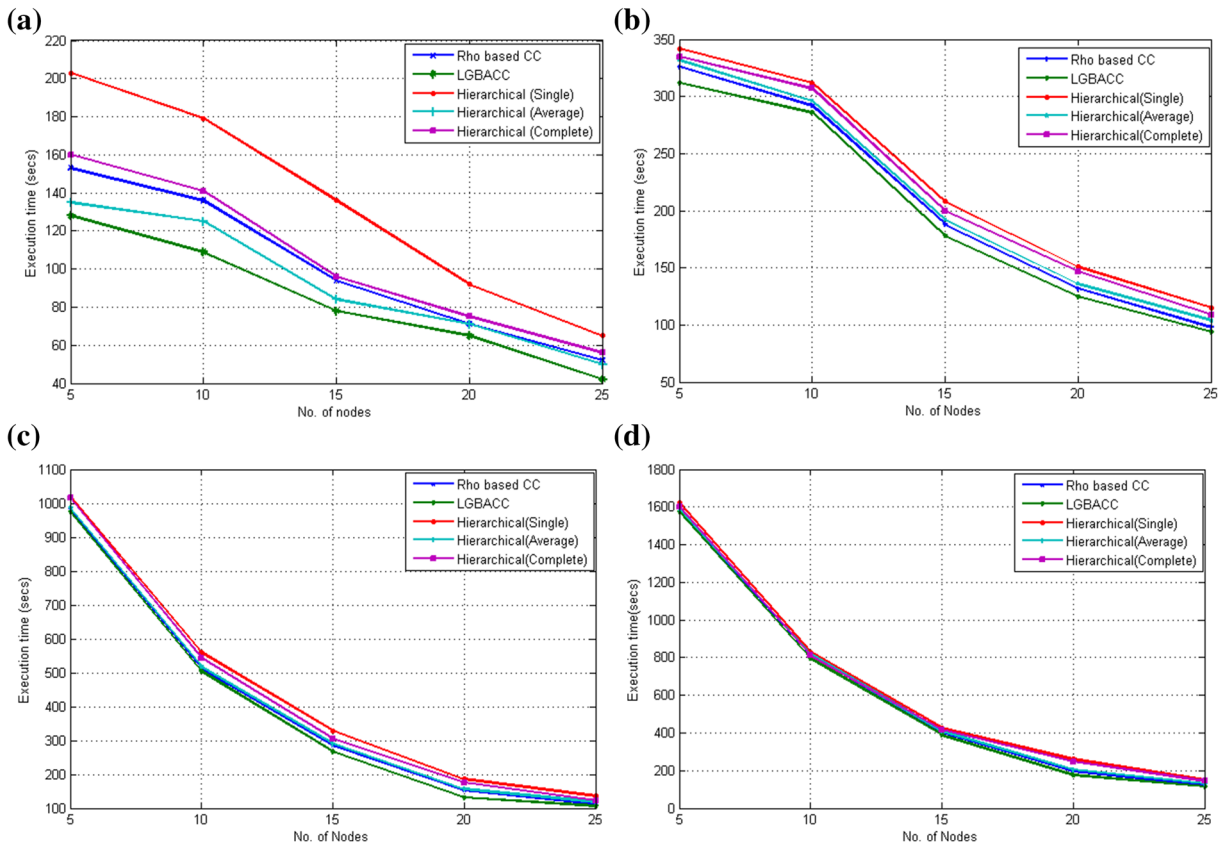

Fig. 10 Comparison of proposed algorithms with existing hierarchical clustering algorithms based on execution time using datasets a) DS1 b) DS2 c) DS3 d) DS4 depicting that the execution time for LGBACC is minimum for all the datasets considered

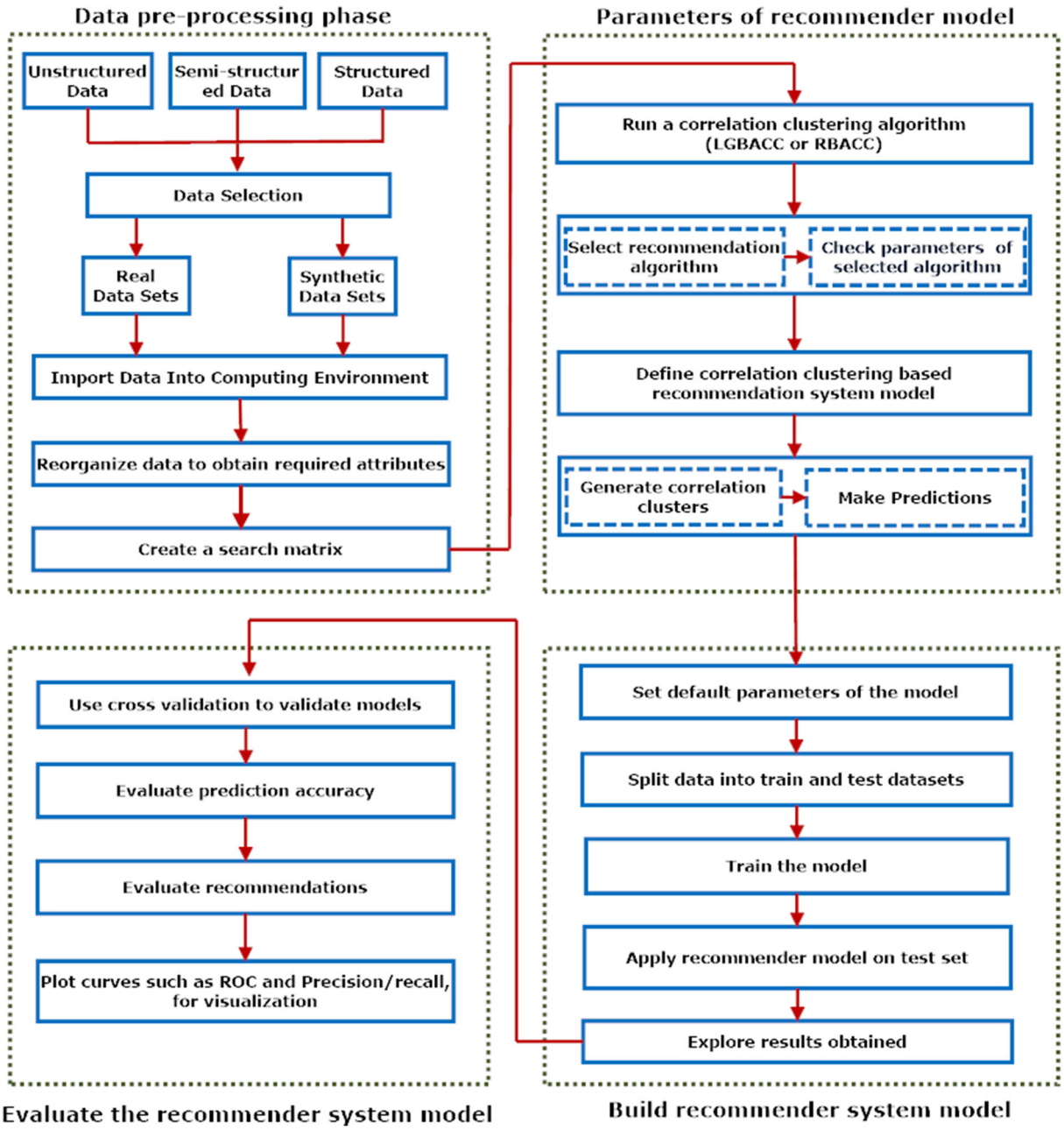

Fig. 11 Framework for the proposed Recommendation System Model

dependency models for items given by some pre-defined association rules.

- k-fold cross validation [32]: The dataset is split into k sets (called folds) of approximately the same size. The evaluation is done k times, using one fold as a test fold and the all the other folds are used for learning. This is a robust approach for evaluating recommender algorithms, making sure that every user is in the test data, at least one. The averaging approach ensures robust results and error estimates.
- Evaluation of predicted values: The best way to evaluate a predicted value is to use measures such as Mean Average Error (MAE), Mean

Square Error (MSE) and Root Mean Square Error (RMSE) [33].

- MAE: This measure helps to evaluate a prediction by computing the deviation of a prediction from the true value. It is calculated as:

$$MAE = \frac{1}{|\kappa|} \sum_{(i,j)\in\kappa} |r_{ij} - \hat{r}_{ij}| \tag{9}$$

where, κ is the set of all user item pairings (i,j), $\hat{r}_{ij}$ is the predicted rating and r_{ij} is the known rating, that was not used to learn the model.

- RMSE: This is another popular measure to find out accuracy. It detects larger errors better than MAE and is suitable to situations where small prediction errors are not very important to find. It can be computed as:

$$RMSE = \sqrt{\frac{\sum_{(i,j)\in\kappa}(r_{ij} - \hat{r}_{ij})^2}{|\kappa|}} \tag{10}$$

MSE = $(RMSE)^2$

- Precision [34]: It is an information retrieval measure, it evaluates recommender performance using:

$$Precision = \frac{Correctly\ recommended\ items}{Total\ recommended\ items} \tag{11}$$

In terms of confusion matrix, given in Table 5, Precision can be expressed as:

$$Precision = \frac{d}{b+d} \tag{12}$$

- Recall [34]: It uses useful recommendations to define a measure for evaluating recommender performance. It is calculated as:

$$Recall = \frac{Correctly\ recommended\ items}{Total\ useful\ recommendations} \tag{13}$$

In terms of confusion matrix, it is represented as:

$$Recall = \frac{d}{c+d} \tag{14}$$

Table 5 Representation of confusion matrix

Actual / Predicted	Negative	Positive
Negative	a	b
Positive	c	d

- ROC (Receiver Operating Character): It is a method to compare two classifiers at different parameter settings. The ROC curve is a method to detect system's probability (also known as sensitivity or true positive rate (TPR)) by the false positive rate, with regard to model parameters. The efficiency of the two systems can be computing by looking at the area under the ROC curve. Bigger area indicates better performance.

6.2 Model Explanation

The phases of the recommender system model presented in Fig. 11 are explained as follows:

- Data pre-processing phase: After the dataset is imported into the computing environment, it needs to be organized in a way that it contains attributes that are required to make recommendations. Once the data is in the desired form, it is converted into a data matrix, known as the search matrix.
- Select parameters of recommender model: The search matrix acts as an input for running correlation clustering algorithms like LGBACC, or RBACC. Once correlation clusters are obtained, a recommendation algorithm is selected depending on the kind of recommendations to be made. The next step is to define the parameters of the selected algorithm. This is done in preparation of defining correlation clustering based recommendation system model. The core of this model is forming a relationship between generated correlated data points and parameters of the recommendation algorithm.
- Build recommender system model: Once the desired data is obtained, and the parameters of the recommendation algorithms are understood, the default parameters of the recommender model

Table 6 Descriptive statistics of movies.csv

MovieId		Title	Genres
Min.	1	Length:40110	Length:40110
1st Qu.	32972	Class :character	Class :character
Mean	86841	Mode :character	Mode :character
3rd Qu.:	134560		
Max.	165201		
Median	98457		

Table 7 Descriptive statistics of ratings.csv

	User ID	Move ID	Rating	Timestamp
Min.	1	1	0.500	7.897e+08
1st Qu.	63930	1015	3.000	9.784e+08
Median	129401	2424	3.500	1.132e+09
Mean	129374	13535	3.527	1.150e+09
3rd Qu.	194037	5816	4.000	1.308e+09
Max.	259137	165201	5.000	1.477e+09

are defined. The next step is to take the clustered dataset and split it into train and test data sets. Then the data is trained and recommender model is applied on the test data set. This is the core of the proposed framework. The last step of this phase is to explore the results obtained.

- Evaluate recommender system model: k-fold cross validation method is used to validate the results obtained in the last phase. The next step is to evaluate the prediction accuracy by using parameters such as RSME, MSE and MAE. Then the recommendations are evaluated by using metrics like Precision and Recall. The last step is to visualize the quality of the recommendations made by plotting curves such as ROC and Precision/Recall.

6.3 Experimental Evaluation

We validate the recommendation system model by using a real world MovieLens dataset, taken from http://grouplens.org/datasets/movielens/latest. This dataset has 24,000,000 ratings and 670,000 tag applications applied to 40,000 movies by 260,000 users. The dataset is approximately 1 Gb is size and has been used by many researchers to make recommendations [35–38]. The latest version of the dataset has information about users from the year 1996 to 2016.

There are four linked files, namely links.csv, movies.csv, ratings.csv and tags.csv. For testing the recommender model, only movies.csv and ratings.csv have been used. The descriptive statistics of the two files are given in Tables 6 and 7. The search matrix is prepared by extracting a list of genres and making each genre a separate attribute. There are 18 genres, combining those with the attributes of 'movies.csv' the search matrix has 20 attributes. We run both LGBACC and RBACC algorithms on this matrix. These algorithms provide similar data points, grouped together. We explore the data further and map the search matrix with 'ratings.csv' to prepare the data for making recommendations. We also normalize this data to make it cleaner for the next step of the recommendation system model. As for the recommendation

Table 8 Evaluation of predicted values with LGBACC having approximately 25% better prediction capability than RBACC but more prediction time

		LGBACC	Rho based CC
Evaluation of predicted values	RMSE	0.9442798	1.2501181
	MSE	0.8916644	1.5627951
	MAE	0.7270231	0.9129824
Model time	Case 1	0.004sec	0.823sec
	Case 2	0.003sec	0.795sec
	Case 3	0.003sec	0.753sec
	Case 4	0.005sec	0.746sec
Prediction time	Case 1	0.175sec	0.033sec
	Case 2	0.166sec	0.03sec
	Case 3	0.18sec	0.031sec
	Case 4	0.17sec	0.031sec

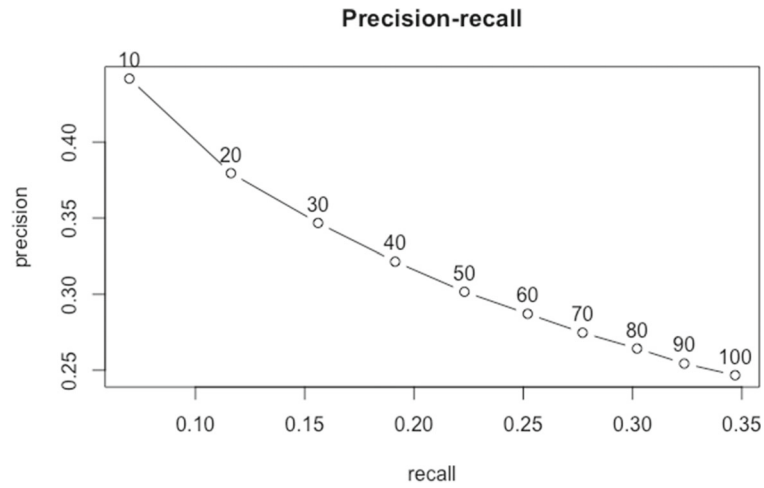

Fig. 12 Precision-Recall curve comparison for LGBACC where UBCF_LGBACC and UBCF_RHO outperforms others

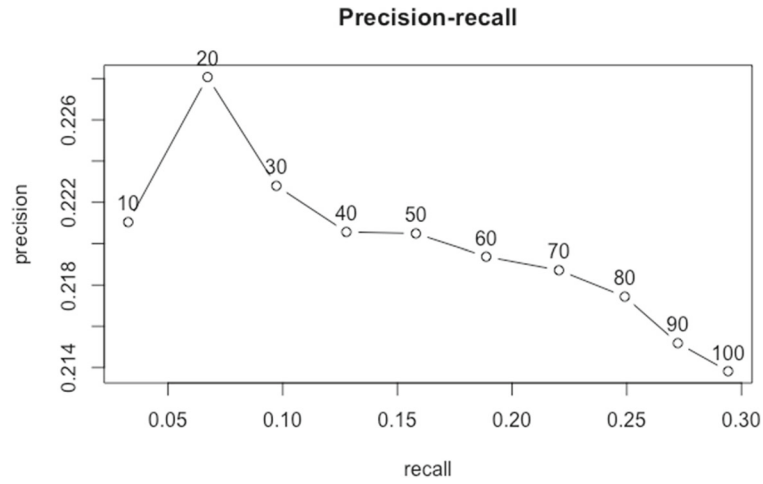

Fig. 13 Precision-Recall curve comparison for RBACC where UBCF_LGBACC and UBCF_RHO have higher ROC than others

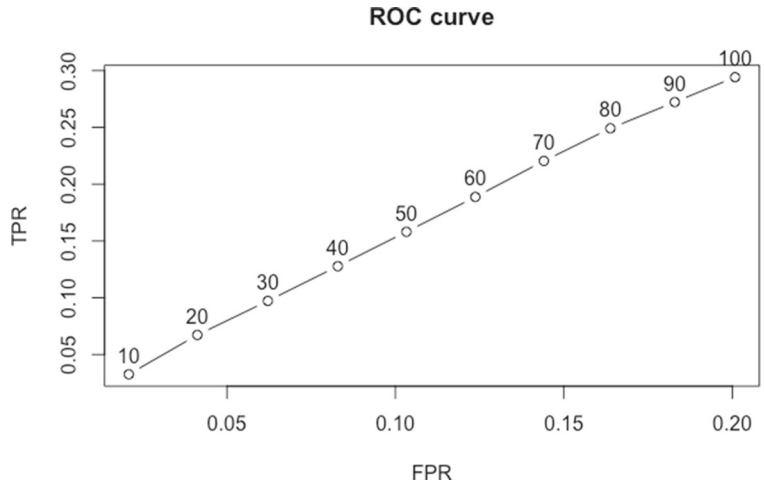

Fig. 14 ROC curve for LGBACC

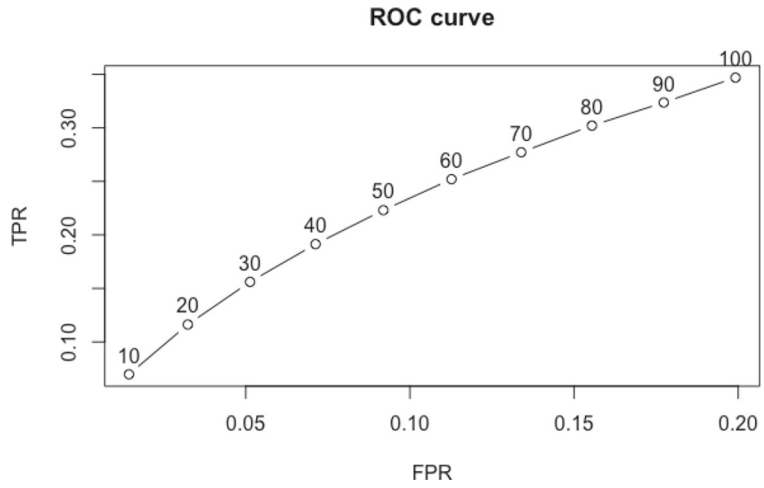

Fig. 15 ROC curve plotted for RBACC

algorithm, we select Item-based collaborative filtering (IBCF) model for our analysis. Looking at the requirements of the algorithm, we determine that the defining parameters of the algorithm are user ratings and preferences. These are the building blocks of the recommender system model. We need to link the correlation clusters with parameters of the recommendation algorithm. Linking correlated data points obtained to the user ratings and preferences, defines the correlation clustering based recommender system

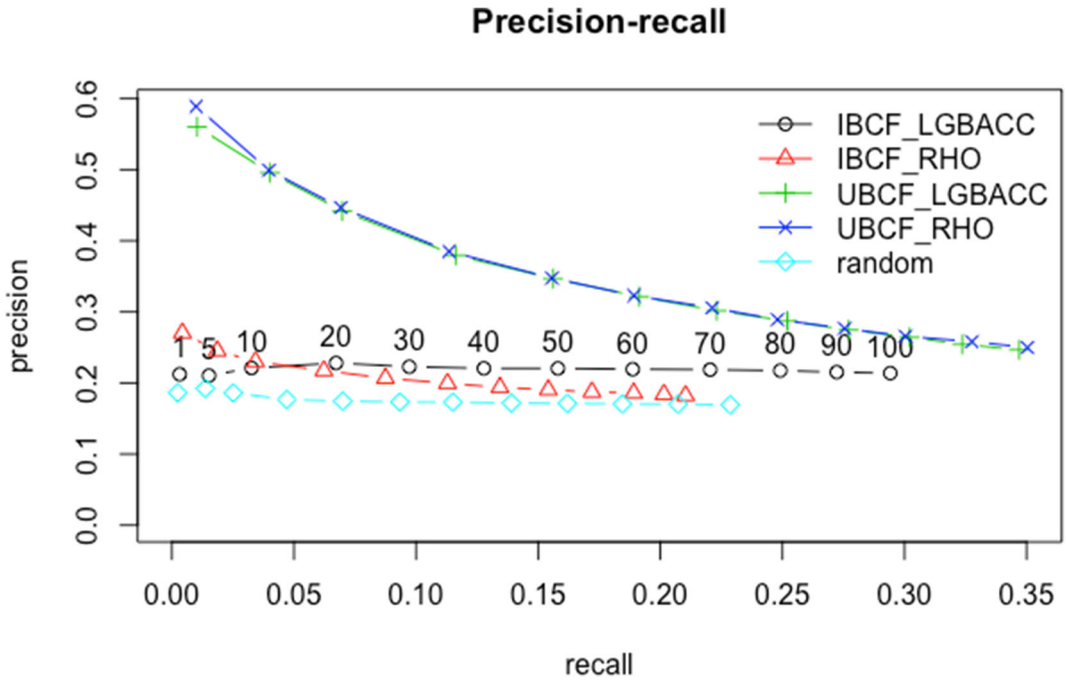

Fig. 16 Algorithms comparison based on Precision-recall

model. Next step is to set the default parameters of the IBCF model. We set the default parameter of method = cosine, and k (the number of items to compute similarities) as 30. Now, we build the model by splitting the whole data set as 80% training data and 20% testing data. Then we apply the recommender system on the dataset, by training and testing the data. We validate the results obtained by running a 4-fold cross validation models. The results of this analysis in terms of RSME, MSE and MAE are given in Table 8. We see that LGBACC has approximately 25% better prediction capability than RBACC. Running this cross validation model for all the folds, we arrive at the conclusion that model time for LGBACC is much lass than that of RBACC, but it takes significantly more prediction time.

Now we evaluate the recommendations by calculating Precision and Recall based on the confusion matrix. The confusion matrix for LGBACC and RBACC are given below:

Rows	**TP**	**FP**	**FN**	**TN**	**precision**	**recall**	**TPR**	**FPR**
10	4.59434	5.216981	71.13208	358.0566	0.4682692	0.07358824	0.07358824	0.01403121
20	7.490566	12.132075	68.23585	351.1415	0.3817308	0.11349283	0.11349283	0.0328015
30	10.084906	19.349057	65.64151	343.9245	0.3426282	0.14647926	0.14647926	0.05233702
40	12.537736	26.707547	63.18868	336.566	0.3194712	0.18124044	0.18124044	0.0723574
50	14.698113	34.358491	61.0283	328.9151	0.2996154	0.21481573	0.21481573	0.09344943
60	16.858491	42.009434	58.86792	321.2642	0.2863782	0.24290723	0.24290723	0.11428793

Rho based CC confusion matrix

Rows	**TP**	**FP**	**FN**	**TN**	**precision**	**recall**	**TPR**	**FPR**
10	2.066038	7.745283	73.66038	355.5283	0.2105769	0.03106647	0.03106647	0.02138576
20	4.396226	15.226415	71.33019	348.0472	0.2240385	0.06719202	0.06719202	0.04181422
30	6.490566	22.943396	69.23585	340.3302	0.2205128	0.09499043	0.09499043	0.0628056
40	8.509434	30.735849	67.21698	332.5377	0.2168269	0.12448159	0.12448159	0.08416257
50	10.566038	38.292453	65.16038	324.9811	0.2157479	0.15244732	0.15244732	0.10485478
60	12.660377	45.650943	63.06604	317.6226	0.2160726	0.17919405	0.17919405	0.1248983

LGBACC confusion matrix

We now plot the ROC and Precision-Recall curves for both LGBACC and RBACC from the confusion matrix. Looking at the Precision-Recall curve Figs. 12 and 13, we see that LGBACC has better Precision-Recall ratio than RBACC. But the area under the ROC curve for LGBACC is less than the other approach as depicted in Figs. 14 and 15.

We compare the existing recommendation models with each other using the IBCF and UBCF models. We create recommendation system models for both LGBACC and RBACC and compare them with the existing random recommendation system model. Precision-Recall and ROC curves are plotted for all the algorithms given in Figs. 16 and 17. It can be

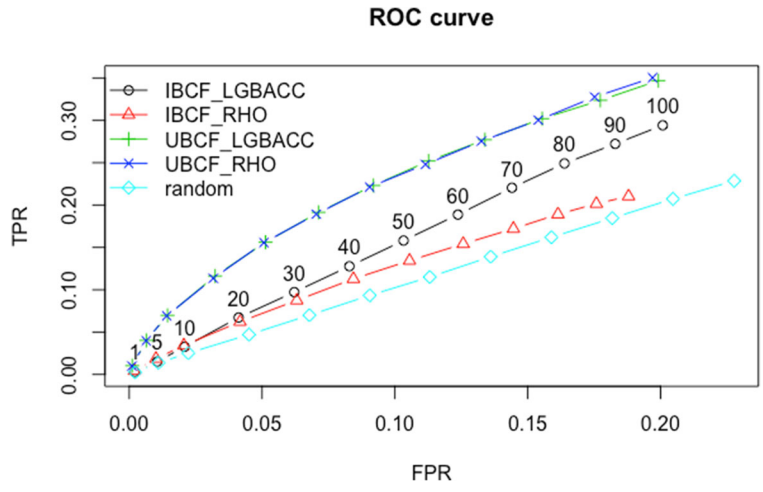

Fig. 17 Algorithms comparison based on ROC

observed that Random recommendation algorithm, that does not use the power of correlations performs the worst, on both the criteria. LGBACC and RBACC using UBCF, perform the best. This analysis shows that integrating correlated points with recommendation algorithms make better quality recommendations.

7 Conclusion

In this paper, we have defined correlation clusters and have proposed two correlation clustering algorithms. These algorithms are similar as well as very distinct. They are similar as both of them exploit the power of correlations in data points and use hierarchical clustering to produce high quality clusters. They are distinct as they are based on totally different concepts of correlation analysis. RBACC algorithm has used the mathematical concept of correlation determination, to produce clusters based on correlation coefficients, calculated using spearman's rank correlation coefficient (ρ). Locality assumption graph based correlation clustering (LGBACC) combines the concepts of dimensionality reduction and graph theory to determine correlations among data objects. Further, it uses hierarchical clustering to produce quality data clusters. Both these algorithms act as the building blocks for designing a recommendation system model through correlations. This model uses correlation clustering algorithms to determine similar data points, use this measure of similarity and merge it with existing recommendation algorithms. It has been discovered that recommendations made using correlated data points are better and faster to compute. Parallel and distributed computing environment has been used to perform all types of analysis in this paper. We have used Spark and R platform to process high-dimensional and large datasets.

Data availability statement (DAS) Data available in a public repository and the links have been provided in relevant section.

References

1. Didelez, V., Pigeot, I.: Judea pearl: Causality: Models, reasoning, and inference. Politische Vierteljahresschrift **42**(2), 313–315 (2001)
2. Reshef, D.N., Reshef, Y.A., Finucane, H.K., Grossman, S.R., McVean, G., Turnbaugh, P.J., Lander, E.S., Mitzenmacher, M., Sabeti, P.C.: Detecting novel associations in large data sets. Science **334**(6062), 1518–1524 (2011)
3. Armanfard, N., Reilly, J.P., Komeili, M.: Local feature selection for data classification. IEEE Trans. Pattern Anal. Mach Intell. **38**(6), 1217–1227 (2016)
4. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Mach. Learn. **56**(1-3), 89–113 (2004)
5. Böhm, C., Kailing, K., Kröger, P., Zimek, A.: Computing clusters of correlation connected objects. In: Proceedings of the 2004 ACM SIGMOD International Conference on Management of data, pp. 455–466. ACM (2004)
6. Aggarwal, C.C., Yu, P.S.: Finding Generalized Projected Clusters in High Dimensional Spaces, vol. 29. ACM, New York (2000)
7. Yang, J., Wang, W., Wang, H., Yu, P.: /spl delta/-clusters: capturing subspace correlation in a large data set. In: Data Engineering, 2002 Proceedings 18th International Conference on, pp. 517–528. IEEE (2002)
8. Achtert, E., Bohm, C., Kroger, P., Zimek, A.: Mining hierarchies of correlation clusters. In: Scientific and Statistical Database Management, 2006 18th International Conference on, pp. 119–128. IEEE (2006)
9. Li, J., Huang, X., Selke, C., Yong, J.: A fast algorithm for finding correlation clusters in noise data. Adv. Know. Discov. Data Min., 639–647 (2007)
10. Achtert, E., Böhm, C., Kriegel, H.-P., Kröger, P., Zimek, A.: Robust, complete, and efficient correlation clustering. In: Proceedings of the 2007 SIAM International Conference on Data Mining, pp. 413–418. SIAM (2007)
11. Achtert, E., Bohm, C., Kriegel, H.-P., Kroger, P., Zimek, A.: On exploring complex relationships of correlation clusters. In: Scientific and Statistical Database Management, 2007 SSBDM'07, 19th International Conference on, pp. 7–7. IEEE (2007)
12. Mukhopadhyay, P., Chaudhuri, B.B.: A survey of hough transform. Pattern Recogn. **48**(3), 993–1010 (2015)
13. Chattopadhyay, A.K., Chattyopadhyay, T., De, T., Mondal, S.: Independent component analysis for dimension reduction classification: Hough transform and cash algorithm. In: Astrostatistical Challenges for the New Astronomy, pp. 185–202. Springer (2013)
14. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications, vol. 27. ACM, New York (1998)

15. Aggarwal, C.C., Wolf, J.L., Yu, P.S., Procopiuc, C., Park, J.S.: Fast algorithms for projected clustering. In: ACM SIGMoD Record, vol. 28, pp. 61–72. ACM (1999)
16. Kailing, K., Kriegel, H.-P., Kröger, P.: Density-connected subspace clustering for high-dimensional data. In: Proceedings of the 2004 SIAM International Conference on Data Mining, pp. 246–256. SIAM (2004)
17. Procopiuc, C.M., Jones, M., Agarwal, P.K., Murali, T.: A monte carlo algorithm for fast projective clustering. In: Proceedings of the 2002 ACM SIGMOD international conference on Management of data, pp. 418–427. ACM (2002)
18. Wang, H., Wang, W., Yang, J., Yu, P.S.: Clustering by pattern similarity in large data sets. In: Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data, pp. 394–405. ACM (2002)
19. Pei, J., Zhang, X., Cho, M., Wang, H., Yu, P.S.: Maple: A fast algorithm for maximal pattern-based clustering. In: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, pp. 259–266. IEEE (2003)
20. Liu, J., Wang, W.: Op-cluster: Clustering by tendency in high dimensional space. In: Data Mining, 2003. ICDM 2003. Third IEEE International Conference on, pp. 187–194. IEEE (2003)
21. Melhem, R.: Parallel gauss-jordan elimination for the solution of dense linear systems. Parallel Comput. **4**(3), 339–343 (1987)
22. Zimek, A.: Correlation clustering. ACM SIGKDD Explor. Newsl. **11**(1), 53–54 (2009)
23. Feng, J., Lin, Z., Xu, H., Yan, S.: Robust subspace segmentation with block-diagonal prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3818–3825 (2014)
24. Kim, Y., Mesbahi, M.: On maximizing the second smallest eigenvalue of a state-dependent graph laplacian. IEEE Trans. Autom. Control **51**(1), 116–120 (2006)
25. Kauffman, L., Rousseeuw, P.: Finding groups in data. An introduction to cluster analysis. John Willey & Sons, New York (1990)
26. Meilă, M.: Comparing clusterings? an information based distance. J. Multiv. Anal. **98**(5), 873–895 (2007)
27. Xiao, C., Ye, J., Esteves, R.M., Rong, C.: Using spearman's correlation coefficients for exploratory data analysis on big dataset. Concurr. Comput. Pract. Experience, 1–13 (2015)
28. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM Sigmod Record, vol. 29, pp. 1–12. ACM (2000)
29. Deshpande, M., Karypis, G.: Item-based top-n recommendation algorithms. ACM Trans. Inf. Syst. (TOIS) **22**(1), 143–177 (2004)
30. Lee, J.-S., Jun, C.-H., Lee, J., Kim, S.: Classification-based collaborative filtering using market basket data. Expert Syst. Appl. **29**(3), 700–704 (2005)
31. Demiriz, A.: Enhancing product recommender systems on sparse binary data. Data Min. Knowl. Disc. **9**(2), 147–170 (2004)
32. Gunawardana, A., Shani, G.: A survey of accuracy evaluation metrics of recommendation tasks. J. M. Learn. Res. **10**, 2935–2962 (2009)
33. Hahsler, M.: recommenderlab: A framework for developing and testing recommendation algorithms, Southern Methodist University (2011)
34. Chowdhury, G.G.: Introduction to modern information retrieval. Facet Publishing, London, United Kingdom (2010)
35. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. (TiiS) **5**(4), 19 (2016)
36. Verbert, K., Drachsler, H., Manouselis, N., Wolpers, M., Vuorikari, R., Duval, E.: Dataset-driven research for improving recommender systems for learning. In: Proceedings of the 1st International Conference on Learning Analytics and Knowledge, pp. 44–53. ACM (2011)
37. Shah, M., Parikh, D., Deshpande, B.: Movie recommendation system employing latent graph features in extremely randomized trees. In: Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, p. 42. ACM (2016)
38. Dooms, S., Bellogín, A., Pessemier, T.D., Martens, L.: A framework for dataset benchmarking and its application to a new movie rating dataset. ACM Tran. Intell. Syst. Technol. (TIST) **7**(3), 41 (2016)