CHAPTER 3

ROBUST ENSEMBLE CO-CLUSTERING ALGORITHM (RECCA) FOR DATA CLUSTERING

3.1. Data Co-clustering

Nowadays available data in the world is being increased enormously due to large number of users. The applications of this type of data is in the field of Bioinformatics, where huge amount of data which is in the form of Deoxyribo Nucleic Acid (DNA) sequences, protein sequences and structures, information on biological pathways, etc. Due to the above reasons there are diverse and scattered sources of biological data.

From the various issues, it is clear that the data integration is considered as complex process and uses large amount of data from various sources. There is a possibility of inconsistencies and redundancies which may occur in database because different databases represent a single attribute of a particular concept with different names. Different attribute values of different sources from same object represent different scaling or encoding (Han and Kamber, 2006) hence conflicts may occur between the data values. Data integration does not provide good clustering results because it is tedious. Hence important information in the process will be discarded. Constrained clustering is used to solve the problem by integrating various data sources without dropping any important information. It is the process of basic clustering which also adds the supplementary information with some constraints that is to be fulfilled by the clustering algorithm. This clustering algorithm solves the problem without the cost of an Apriori data integration process using the really valuable information.

The major objective of this algorithm is to determine data integration in the form of complete datasets, which might give vital information regarding

the functions and data structures, as well as functional diversification developed during evolution of family.

The enhanced principal component analysis is proposed in this research work. Then the objective function for the application of enzyme clustering is presented. Constructive mathematical modeling is used to describe spectral co-clustering ensemble algorithm with brief description. The proposed algorithm is used to do co-clustering with the objective function as the principal component.

3.2. Missing Values Problem

Real world data are often having missing values. The quality of the supervised learning process is affected by bias which is created by missing values. In machine learning and data mining, the quality of the data is the keypoint. The missing values are found by an efficient way which is missing value imputation that is based on other information in the datasets.

Based on Little and Rubin (2014) different data mechanisms are classified into three major categories.

Missing Completely At Random (MCAR) : The level of randomness is high in MCAR. There is no other reason behind the missed data. If any missing variable X is independent, it mightn't calculate the missing variable X from remaining variables in dataset. So, all the missing variables are related to the probability of the missing variables. The known values or the missing values does not relate to any attribute. The missing data that occurs appropriate to structural reasons might not be considered as MCAR (London School of Hygiene and Tropical Medicine, 2013).

Missing At Random (MAR): This is entirely different from the MCAR. In the given dataset the values on missing variable X are predicted based on the other variable Y, but does not rely on the missing data value itself. For example consider food consumption for analyzing the food(X) and weight(Y) relationship. Based on the food taken, the weight of food will

increase or decrease. So the relation among food and weight is correct example of MAR. Within each class, MAR is uniform and non-response. Consider an example (London School of Hygiene and Tropical Medicine, 2013) to collect an income and property tax band data. Normally, higher income values data are less and that are willing to reveal them. If a property tax band data is non-response, then the income data is missed randomly. The missed data completely depends on non-response that is property tax band. But non-response can't depend on income itself.

Not Missing At Random (NMAR): The missing variable cannot be predicted from other variables in the dataset and it is not random. Some sources are referred to this mechanism as Missing Not At Random (MNAR) (Horton and Kleinman, 2012).

3.2.1. Treatment of Missing Data

1. Ignoring and removing data: In each instance, these methods determine the missing attributes and those instances are deleted. Next each attribute/instances are determined and whole attributes are removed which is having high level of missing data. If the dataset belongs to MCAR then this is valid.

2. Parameter estimation: The parameters for the complete data are found using this method. For example Expectation-maximization algorithm based parameter estimation algorithm is mostly used for the parameter estimation of the missing data.

3. Imputation technique: Based on the estimated values the missing values are replaced by this method.

3.2.2. Methods for Handling with Missing Data Problem

There are three main approaches in handling with missing data. The missing values imputation problem provides solution for reduction of the data set and discarding of all samples with missing values (Kantardzic, 2003). Additional solution is the consideration of missing values as special values.

Lastly different missing values imputation methods solve many missing values problem. Inappropriately missing values imputation methods are appropriate only for missing values produced by MCAR and some methods for MAR mechanism. If missing values are initiated by NMAR, the initial data must be analyzed and from the analysis, suitable model for the missing data can be obtained.

3.2.2.1. Analytical Method

If a method and its policy have been used already for data analysis particularly in missing data then there is no need to use a particular method for missing values. Attributes with missing values are considered as irrelevant (Luengo, 2011) in decision rules extraction methods. Some extraction methods with association rules eliminate missing values in the rows (conservative approach) or missing values which handle the rules in the method (optimistic approach).

3.2.2.2. Reducing the Data Set

Dataset reduction and missing value elimination are the simplest solution for solving missing data problem. This is achieved by the elimination of missing values in samples (rows) (Horton and Kleinman, 2012) or elimination of missing values in the attributes (columns) (Lakshminarayan et al., 1999). Elimination of all samples which means combining both schemas is also called as complete case examination. All samples can be eliminated only when huge data sets are obtainable, and only a small percentage of missing values occur in samples. Complete example examination will not lead to severe problem during the inference. Attribute elimination with missing values through examination does not provide better results because it eliminates most of the important data. It is important to make inferences about these attributes. Both approaches decrease the information content of the data so they are useless procedures.

3.2.2.3. Special Values

Completely different approach is used to remove the unknown attribute values in this method. Instead of finding some known attribute value as its value, finding missing value itself as a new attribute values will provide better results and it is like similar as other attribute values (Grzymala-Busse, and Hu, 2001). Storing missing value information is better than storing attribute value. This approach is used to handle values without affecting the future analyses.

3.2.2.4. Mean

In this method, the missing values are replaced by mean value of the attribute "(London School of Hygiene and Tropical Medicine, 2013)". All known values of the attribute are used to calculate the mean value. It is suitable for numeric attributes and is joined with missing values among the most frequent attribute value for symbolic attributes.

Rahman and Islam (2016) introduce a new Fuzzy Expectation Maximization and Fuzzy clustering methods for missing value (FEMI) by computing mean value of the attribute.

3.2.2.5. Mean for the Given Class

This method is related to the mean method. The difference is that the mean value is not computed from all known attribute values, but it is calculated based on the attributes values belonging to specified class. This method is suitable for only classification related problems where samples are classified already (London School of Hygiene and Tropical Medicine, 2013) or it is possible to create the classes.

3.2.2.6. Median for the Given Class

Meanwhile the occurrence of outliers affects the mean of attributes it is usual to utilize the median value instead of guaranteeing robustness. In this case the particular attribute, which is missing is replaced by the median of all

known attribute values in the class (Acuña and Rodriguez (2004)). This method is suitable only for numeric attributes and needs presence of classes.

3.2.2.7. Most Common Attribute Value

In this schema most frequent attribute values are used for missing value data problem (Eskelson *et al.*, 2009). This schema is mostly used for symbolic attributes and replacing missing values using mean value of numerical attributes.

3.2.2.8. Concept Most Common Attribute Value

The procedure of this schema is similar like as previous schema (Eskelson *et al.*, 2009). This schema involves most common attribute value but involves some values that are belonging to the given class. This schema is applicable to symbolic attributes and need class existence.

3.2.2.9. k-Nearest Neighbor(KNN)

K Nearest Neighbor (KNN) generally searches more than one ("k") related cases with known attribute values. The objects are classified using nearest neighbor methods. A new object is classified with input vector y via examining the "k" closest data set points to y and assigns the object to the class that has highest similarity among these "k". Numeric attributes are used to calculate a weighted mean of the KNN attribute values (Hu, 2003). The weights are inversely proportional to the distances between neighboring columns. "k" parameter selection is the main problem. Another problem is handling symbolic attributes. Distance is measured by numerous metrics. Some the distance metrics are Euclidean distance, Manhattan distance and many others.

3.2.2.10. Neural Networks

Neural Networks (NN) comprises of a class of predictive modeling system and it is worked based on the iterative parameter adjustment (Bishop, 2006). The structure of NN is called as architecture, which consists of the

number of neurons, number of layers, network model type, etc., and the interconnection structure. Single-layer network contains only input and output layer. A multilayer network contains more hidden layers that are intermediate between the input and the output layer. In this research work, it is clear from all of these categories missing Values are replaced with Mean value. The mean of the attribute is calculated based on all known attribute values which are explained in this research work.

Ji et al (2015) proposed a new Radial Basis Function (RBF) network to find the missing values of the user-item rating matrix.

3.3. General Procedure of PCA for Missing Data Problem

Data sets in real time are having many missing values. Solving this problem and procedures are considered on a common level. If there is less number of missing values, the usual method will skip the computations of the data vectors in which some of the components are missing. But if the data has a major amount of missing values, this simple procedures lose more information. Another simple procedure is to calculate the sample co-variance matrix using component pairs of the data vectors for which both components are not missing. On the other hand, this method provides an invalid sample co-variance matrix that is no longer positive (semi)definite (Ilin and Raiko, 2000).

While the PCA problem with missing values may be considered as easy as normal PCA, there are some significant differences (Ilin and Raiko, 2000). Due to these reasons this situation is considered as nonlinear modeling problems. For missing values there is no systematic solution presented for PCA since the evaluation of the covariance matrix is nontrivial. Following, the optimized cost function logically has local minima, so it becomes very difficult to determine the global optimum value.

Probabilistic approaches use regularization concept because standard PCA approaches will provide over fitting. In large-scale problems, this

algorithm provides a heavy computational load. All the methods which are considered here assume that missing values in each part of a data vector created at random. In the same way, outliers created in each component of a data vector randomly. Opposing to this, at least (Archambeau *et al.,* 2006) entire data vectors are considered to be outliers in some probabilistic methods. A correct model should be used for the existence of missing values and evaluate the input samples for outliers.

A usual method used to handle missing values is so-called imputation method (Luttinen *et al.*, 2009). Here the PCA method replaces the missing data by computing the mean value of that component. Since in missing data imputation zero mean will be used, in this work the missing values are replaced by zeros. The sample co-variance matrix can be estimated and its PCA can also be computed in the normal manner. A more advanced PCA methods uses iterations for replacing the missing data values.

Inspite of PCA having attractive features, its models has the several shortcomings. The principal component directions are found by naive methods that have some trouble with high dimensional data or large numbers of data points. The calculation of sample co-variance matrix of vectors in a space of several hundred or several thousand dimensions is difficult. These difficulties are computational complexity, data scarcity and high cost. It is clear that it is better to avoid the sample co-variance matrix computation. This problem in this work is solved by proposing a new Enhanced Principal Component Analysis (EPCA) that uses a new co-variance matrix value.

3.4. Enhanced Principal Component Analysis

An Enhanced Weighted version of PCA (EPCA) is introduced where most significance is known to observations whose values are more significant. Here the enhanced PCA uses a Pearson's correlation coefficient which gives higher weights to observations which are considered to be more important.

Also, the Pearson's correlation coefficient is perceptive to the occurrence of outliers and noise in the data. The ranks of the observations are used. In the meeting dataset ranking the observations for each conversation from 1 (highest rank) to n (lowest rank) is taken. This coefficient of the ranked data is thus obtained using the Spearman's rank correlation coefficient r_s, which is specified by the expression is found in "Uma and Suguna, 2015" as follows,

$$r_s = \frac{\sum_{i=1}^{n}(R_i - R)(Q_i - Q)}{\sqrt{\sum_{i=1}^{n}(R_i - R)^2 \sum_{i=1}^{n}(Q_i - Q)^2}} \tag{3.1}$$

where R and Q are the average ranks. However to reduce computational cost the expression "Uma and Suguna, 2015" which assumes there are no ties.

$$r_s = 1 - \frac{6\sum_{i=1}^{n}(R_i - Q_i)^2}{n^3 - n} \tag{3.2}$$

It is clear from this rewritten form of r_S with the purpose of the computation of the distance among two ranks in Spearman's coefficient is specified "Uma and Suguna, 2015" by

$$D_i^2 = (R_i - Q_i)^2 \tag{3.3}$$

which it does not take rank importance into account, because if $(R_i - Q_i)$ is, for instance, (1, 3) or (n- 2,n), the contribution is the same. The following different distance measure "Uma and Suguna, 2015" is proposed:

$$WD_i^2 = (R_i - Q_i)^2\big((n - R_i + 1) + (n - Q_i + 1)\big) \tag{3.4}$$

$$WD_i^2 = D_i^2(2n + 2 - R_i - Q_i) \tag{3.5}$$

The first term of this product is D_i^2, accurately as in Spearman's coefficient, and represents the distance between R_i and Q_i; the second term is a linear weighting function which represents both the significance of R_i and Q_i. Hence the weighted rank measure of correlation "Uma and Suguna, 2015" is obtained using

$$r_w = 1 - \frac{6\sum_{i=1}^{n}(R_i - Q_i)^2\,(2n + 2 - R_i - Q_i}{n^4 + n^3 - n^2 - n} \tag{3.6}$$

which yields the values between -1 and +1. The calculation of the distance between two ranks R_i and $Q_i$is given by $WD_i^2 = (R_i - Q_i)^2(2n + 2 - R_i - Q_i)$ where the second term of the product is a linear weighting function which represents the importance of R_i and Q_i. Hence, the distance measure is

$$W_2D_i^2 = (R_i - Q_i)^2(2n + 2 - R_i - Q_i)^2 \tag{3.7}$$

which reflects more than WD_i^2 the higher importance of agreement on top ranks. It is common to define rank correlation coefficients, such as Spearman's, as a linear function of the distance between the two vectors of ranks. In this research, this corresponds to define a coefficient of the form

$$W_2D_i^2 = A + B\sum_{i=1}^{n}(R_i - Q_i)^2(2n + 2 - R_i - Q_i)^2 \tag{3.8}$$

where the conversations are such that it takes values between -1 and +1. In order to find A and B, we will start by doing a specific data transformation and then compute the Pearson's coefficient on the transformed data. The expression obtained is exactly of the form, from where the constants A and B follow. The transformation consists in substituting the value of observation i in the first variable by the value $R_i^{'} = R_i(2n + 2 - R_i)$, where R_i is the rank of that observation. It is clear from above, that the computation of the new correlation coefficient is equivalent to do a data transformation to each variable as $R_i^{'} = R_i(2n + 2 - R_i)$ and then compute the Pearson's correlation coefficient. R_i represents the rank of each observation value; usually the smallest value has rank 1, the second smallest rank 2, and so on "Uma and Suguna, 2015".

3.5. General Knowledge of Clustering

After missing data is replaced then clustering problem is focused of primary importance in data clustering. Over the past several decades variety of

methods have been proposed to solve clustering problems (Jain et al., 1999). A comparatively recent area of focus has been *spectral clustering*, a class of methods based on eigen decompositions of affinity, dissimilarity or kernel matrices. They often yield better performance when compared to other clustering algorithms such as k-means, and they have been effectively deployed in various applications such as computer vision, bioinformatics, and robotics. Moreover, there is a substantial theoretical literature supporting spectral clustering (Kannan *et al.,* 2004).

Despite these virtues, spectral clustering is not widely viewed as a competitor to classical algorithms such as hierarchical clustering and k-means for large-scale data mining problems. The reason is easy to state given a data set consisting of n data points, spectral clustering algorithms form an n × n affinity matrix and compute eigenvectors of this matrix, an operation that has a computational complexity of $O(n^3)$ in general. For applications with n on the order of thousands, spectral clustering methods begin to become infeasible, and problems with n in the millions are entirely out of reach.

To solve this spectral clustering problem then co-clustering Ensemble is focused in this research work.

3.6. Co-clustering Ensemble using Spectral Clustering

Given t partitions, with the q^{th} partition $\left(\mu^{(q)}, v^{(q)}\right)$ having $k^{(q)}$ row clusters and $\ell^{(q)}$column clusters. T is defined as a consensus function $N^{\{m\times t, n\times t\}} \rightarrow N^{\{m,n\}}$ mapping a set of co-clusterings to an integrated co-clustering:

$$T: \left\{\left(\mu^{(q)}, v^{(q)}\right) | q \in \{1, \dots, t\}\right\} \rightarrow \{(\mu, v)\} \tag{3.9}$$

Let the set of partitions $\left\{\left(\mu^{(q)}, v^{(q)}\right) | q \in \{1, \dots, t\}\right\}$ be denoted by ϕ. If there is no background information regarding the relative value of the individual partitions, then a reasonable objective for the consensus solution is to

search for a co-clustering with the purpose of shares the most information by means of the original co-clusterings.

In order to compute "the statistical information shared between two co-clusterings, mutual information Strehl and Ghosh, 2003 is used as a symmetric measure". Here, the objective function is proposed by adjusting the original definition to handle the problem of co-clustering ensemble:

$$(\mu, v)^{(k,l,-opt)} = \arg\max \sum_{q=1}^{t} \phi^{(NMI)}\{(\mu, v), \mu^{(q)}, v^{(q)}\} \tag{3.10}$$

where $(\mu, v)^{(k,l,-opt)}$ is the optimal combined co-clustering and it is the one that has maximal average mutual information with all individual partitions in ϕ given that the number of consensus row clusters desired is k and the number of column clusters is l. In detail, the Average Normalized Mutual Information $(ANMI)$ Strehl and Ghosh, 2003, between a single co-clustering (μ, v) and a set of t co-clusterings can be defined as

$$\phi^{(ANMI)}\big(\phi, (\mu, v)\big) = \frac{1}{t}\sum_{q=1}^{t} \phi^{(NMI)}\left((\mu, v), \mu^{(q)}, v^{(q)}\right) \tag{3.11}$$

In eqs.(3.11) Normalized Mutual Information (NMI) is defined as

$$NMI(X, Y) = \frac{I(X, Y)}{\sqrt{H(X)H(Y)}} \tag{3.12}$$

where X and Y denote two vectors, $I(X, Y)$ denotes the mutual information between X and Y. $H(X)$denotes the entropy of X and $H(X) = I(X, X)$. Suppose there are two co-clusterings $(X_r, X_c)\ and\ (Y_r, Y_c)$, $i.e., (X_r, Y_r), (X_c, Y_c)$ denote the row and column cluster labeling variables respectively. Then, the NMI among two co-clusterings might be defined as

$$\begin{aligned} NMI\big((X_r, X_c), (Y_r, Y_c)\big) &= NMI\big((X_r, Y_r)\big) + NMI\big((X_c, Y_c)\big) \\ &= \frac{I(X_r, Y_r)}{\sqrt{H(X_r)H(Y_r)}} + \frac{I(X_c, Y_c)}{\sqrt{H(X_c)H(Y_c)}} \end{aligned} \tag{3.13}$$

It is clear that $NMI(X_r, X_r) = NMI\ (Y_c, Y_c) = 1$, as desired. According to Eqs. (3.13), (3.11) can be further rewritten as

$$\phi^{(ANMI)}\big(\phi, (\mu, v)\big) = \frac{1}{t}\sum_{q=1}^{t} \phi^{(NMI)}\left((\mu, v), \mu^{(q)}, v^{(q)}\right) \tag{3.14}$$

$$= \frac{1}{t}\sum_{q=1}^{t}\left(\frac{I\big(\mu, \mu^{(q)}\big)}{\sqrt{H(\mu)H(\mu^{(q)})}} + \frac{I\big(v, v^{(q)}\big)}{\sqrt{H(v)H(v^{(q)})}}\right)$$

Eq. (3.14) needs to be calculated by the sampled quantities given through the co-clusterings. Then, the normalized mutual information approximation $\phi^{(NMI)}$can be defined as,

$$\phi^{(NMI)}\left((\mu^i, v^i)\ , (\mu^j, v^j)\right) \tag{3.15}$$

$$= \phi^{(NMI)}\left((\mu^i, \mu^j)\ \right) + \phi^{(NMI)}\left((v^i, v^j)\right)$$

$$= \frac{\sum_{\alpha=1}^{k(i)}\sum_{\beta=1}^{k(j)} O_{\alpha,\beta}\log\left(\frac{|O|.O_{\alpha,\beta}}{O_\alpha^i O_\beta^j}\right)}{\sqrt{\left(\sum_{\alpha=1}^{k(i)} O_\alpha^i\ log\frac{O_\alpha^i}{|O|}\right)\left(\sum_{\beta=1}^{k(j)} O_\beta^i\ log\frac{O_\beta^i}{|O|}\right)}} +$$

$$\frac{\sum_{\alpha=1}^{\ell(i)}\sum_{\beta=1}^{\ell(j)} O_{\alpha,\beta}\log\left(\frac{|F|.F_{\alpha,\beta}}{F_\alpha^i F_\beta^j}\right)}{\sqrt{\left(\sum_{\alpha=1}^{\ell(i)} F_\alpha^i\ log\frac{F_\alpha^i}{|F|}\right)\left(\sum_{\beta=1}^{\ell(j)} F_\beta^i\ log\frac{F_\beta^i}{|F|}\right)}}$$

where $|O|$ and $|F|$ denote the number of objects and features in a cocluster correspondingly. (O_α^i, F_α^i) denotes the number of objects and features in co-cluster CO_α according to (μ^i, v^i) and (O_α^j, F_α^j) denotes the number of objects and features in co-cluster CO_β according to (μ^j, v^j).

3.6.1. Spectral Co-Clustering Ensemble Algorithm

In this work, the final ensemble step can be formulated as a partition problem on a bipartite graph. For convenience of discussion, use small-bold

letters such as u, v as vectors. Capital-bold letters such as M, E, L will denote matrices, and capital letters such as V, R will denote vertex sets. Denote the bipartite graph $G = (V_r, V_c, E)$ containing two sets of vertices including row labeling vertices V_r and column labeling vertices V_c respectively. It is easy to verify that the adjacency matrix M of the bipartite graph can be written as

$$M = \begin{bmatrix} O & E \\ E^T & O \end{bmatrix} \tag{3.16}$$

Where

$$E = \begin{bmatrix} C_{rr} & C_{rc} \\ C_{cr} & C_{cc} \end{bmatrix} \tag{3.17}$$

C_{rr} denotes the edge-weights between row labeling vertices that are both in V_r . C_{rc} denotes the edge-weights between labeling vertices with one in V_r and the other in V_c . C_{cc}; C_{cr} are defined similarly and $C_{rc} = C_{cr}^T$. Let $|E|_{ij}$ is the edge weight between two vertices and can be obtained from Eq.(3.17).

$$|E|_{ij} = \frac{\sum_{\alpha=1}^{k(i)} \sum_{\beta=1}^{k(j)} O_{\alpha,\beta} \log\left(\frac{|O|.O_{\alpha,\beta}}{O_\alpha^i O_\beta^j}\right)}{\sqrt{\left(\sum_{\alpha=1}^{k(i)} O_\alpha^i \ log \frac{O_\alpha^i}{|O|}\right)\left(\sum_{\beta=1}^{k(j)} O_\beta^i \ log \frac{O_\beta^i}{|O|}\right)}} \tag{3.18}$$

if the i[th] and j[th] vertices are both the row labeling vertices for data clusters “Dhillon , 2001”;

$$|E|_{ij} = \frac{\sum_{\alpha=1}^{\ell(i)} \sum_{\beta=1}^{\ell(j)} O_{\alpha,\beta} \log\left(\frac{|F|.F_{\alpha,\beta}}{F_\alpha^i F_\beta^j}\right)}{\sqrt{\left(\sum_{\alpha=1}^{\ell(i)} F_\alpha^i \ log \frac{F_\alpha^i}{|F|}\right)\left(\sum_{\beta=1}^{\ell(j)} F_\beta^i \ log \frac{F_\beta^i}{|F|}\right)}} \tag{3.19}$$

if the i[th] and j[th] vertices are both the column labeling vertices for data clusters. Otherwise $|E|_{ij} = 0$. According to the bipartite graph $G = (V_r, V_c, \boldsymbol{E})$ given above, now we define the co-clustering partition matrix Y as

$$Y = \begin{bmatrix} Y_r \\ Y_c \end{bmatrix} \tag{3.20}$$

where $\boldsymbol{Y_r}$ is the partition on row labeling vertex set V_r and $\boldsymbol{Y_c}$ is the partition on column labeling vertex set V_c. Thus, the laplacian matrix L can be defined as

$$L = D - M \tag{3.21}$$

where

$$D = \begin{bmatrix} D_r & O \\ O & D_c \end{bmatrix} \tag{3.22}$$

D_r and $D_c$are diagonal matrices such that $|D_r|_{ii} = \sum_j E_{ij}$, $|D_c|_{jj} = \sum_i E_{ij}$. Note that the key step is to find the minimum cut vertex partitions on the bipartite graph. The normalized-cut objective function can be expressed as

$$\min_{\boldsymbol{Y}} \boldsymbol{tr}(\boldsymbol{Y^T L Y}) \tag{3.23}$$

One way to solve the partition problem of the bipartite graph "Dhillon, 2001" is to compute the left and right eigenvectors of the matrix A defined as

$$A = D_r^{-1/2} E D_c^{-1/2} \tag{3.24}$$

After the left and right eigenvectors of matrix **A** are obtained, the left and right eigenvectors of the second to the $(\omega + 1)th$ eigenvalues are selected as $U = [u_2, u_3, \dots, u_{\omega+1}]$ and $V = [v_2, v_3, \dots, v_{\omega+1}]$ respectively. Here, the $\omega = log_2 k$ singular vectors $u_2, u_3, \dots, u_{\omega+1}$, and $v_2, v_3, \dots, v_{\omega+1}$ often contain k-modal information about the original co-clustering labeling. Thus, the k-dimensional data matrix can be written as

$$X = \begin{bmatrix} D_r^{-1/2} & U \\ D_c^{-1/2} & V \end{bmatrix} \tag{3.25}$$

At last, the traditional k-means algorithm is preformed on x-dimensional data, and the final consensus co-clustering result is obtained.

Algorithm Description

The algorithm procedure is described as follows.

Algorithm 3.1.(SPECTRAL CO-CLUSTERING ENSEMBLE)

Input: $Original\ data\ matrix\ X_{mn}, num.of\ row\ clusters\ k,$

$$num.of\ column\ clusters\ \ell\ (i.e., K\ \times\ \ell\ co-clusters\ in\ total)$$

01. Divide X_{mn} into k row clusters and ℓ column clusters.
02. Compute pairwise similarities according to Eqs. (3.12) and (3.13). Create the adjacency matrices **M.**
03. Construct the diagonal matrices D_r, D_c where $|D_r|_{ii} = \sum_j E_{ij}$ and $|D_c|_{jj} = \sum_i E_{ij}$
04. Calculate **A** as defined in Eq. (3.18).
05. Perform Singular Value Decomposition (SVD) on matrix **A**. Compute $\omega = log_2 k$ singular vectors of **A,** $\boldsymbol{u_2}, \ldots \boldsymbol{u}_{\omega+1}$and $\boldsymbol{v_2}, \ldots \boldsymbol{v}_{\omega+1}$. Represent the left and right eigenvectors of the 2nd to the $(\omega + 1)th$ eigenvalues as U and V respectively.
06. Construct $X_r = D_r^{-1/2} U$ and $X_c = D_c^{-1/2} V$.
07. Execute k-means algorithm to x-dimensional data $X_r$toward obtain the row labelings partition matrix Y_r; In the same way get Y_c from X_c .

Output:

$$The\ final\ consensus\ co-clustering\ result.$$

In SVD used a Lanczos algorithm to calculate the eigenvectors "Shi *et al.*, 2010". The complexity of this algorithm is $O(eN(|m| + |n|)^2$, where e is the number of eigenvectors desired, N is the number of iteration steps and $(|m| + |n|)^2$ is the upper bound of the nonzero entries in matrix M.

3.7. Dataset Description

Several datasets have been taken for the performance analysis. The datasets for text pairwise co-clustering is shown in Table 3.1. The datasets for Text High-Order (Word-Document-Category) co-clustering is presented in Table 3.2. The datasets for gene expression pairwise (Condition-Gene) co-clustering is given in Table 3.3. The datasets for Image High-Order (Color-Image-Texture) co-clustering is depicted in Table 3.4.

Samples of oh5 and oh15 are from OHSUMED collection, a subset of MEDLINE database, which consists of 233,445 documents, labeled using 14,321 unique categories. Samples WAP is from the WebACE Project, and every document relates to a web page indexed in the topic hierarchy of Yahoo. Data set re0 is the Reuters-21578 text classification collection (distribution 1.0). Also use the Newsgroup data which consists of 2,000 articles from 20 newsgroups.

(i) Condition-Gene

In this work make use of seven data sets collected from Kent Ridge Biomedical Data Repository (http://datam.i2r.a-star.edu.sg/datasets/krbd/) designed for gene expression co-clustering, with ALL/AML Leukemia, Breast Cancer, Central Nervous System, Colon Tumor, Lung Cancer, Ovarian Cancer, and ALL/MLL/ AML Leukemia sort all genes in a descending order based on the variances and retain simply the top 2,000 genes.

(ii) Image Co-clustering

The image data used in the experiments is selected from Corel CDs, which contains 31,438 wide-ranging-purpose images of a variety of contents, such as plants, animals, buildings, human society, etc. To measure proposed algorithm, build a dataset with 1,000 images from 10 categories: “eggs,” “decoys,” “firearms,” “cards,” “buses,” “abstract,” “foliage,” “dawn,” “texture,” and “wave.”

Table 3.1. Data Sets for Text Pairwise (Document-Word) Co-clustering

Name	Datasets	Data Structure	No. of clusters	No. of documents
CT1	oh15	Adenosine-Diphosphate, Blood-Vessels	2	154
CT2	oh15	Aluminium, Blood-Coagulation-Factors	2	122
CT3	re0	Interest, reserves	2	261
CT4	re0	housing, jobs	2	55
CT5	re0	housing, interest, jobs	3	274
CT6	oh15	Aluminium, Blood-Vessels, Leucine	3	207
CT7	re0	cpi, housing, ipi, lei, retail	5	144
CT8	re0	bop, cpi, gnp, housing, interest, ipi, jobs, lei, money	10	1150

1. http://www.cs.umn.edu/~han/data/tmdata.tar.gz.
2. http://www.cs.uiuc.edu/homes/dengcai2/Data/TextData.html,

Table 3.2. Data Sets for Text High-Order (Word-Document-Category) Co-clustering

Name	Datasets	Data Structure	No. of clusters	No. of documents
HT1	oh15, re0	{ Adenosine-Diphosphate, Aluminium, Cell-Movement}, {cpi,money}	5	899
HT2	oh15, re0	{Blood-Coagulation-Factors, Enzyme-Activation, Staphylococcal-Infections}, {jobs,reserves}	5	461
HT3	oh15, re0	{Aluminium, Blood-Coagulation-Factors, Blood-Vessels}, {housing,retail}	5	256
HT4	oh15, re0	{Aluminum, Cell-Movement, Staphylococcal-Infections}, {cpi, jobs}	5	391
HT5	WAP, re0	{media, film, music}, {cpi, jobs}	5	404

HT6	Newsgroup	{rec.sport.baseball, rec.sport.hockey}, {talk.politics.guns, talk.politics.mideast,talk.politics}	5	500
HT7	Newsgroup	{comp.graphics, comp.os.ms-windows.misc}, {rec.autos,rec.motorcycles}, {sci.encrypt, sci.electronics}	6	300
HT8	Newsgroup	{ comp.graphics, comp.os.ms-windows.misc}, {sci.electronics, sci.med}	4	3932
HT9	Newsgroup	{rec.autos, rec.motorcycles, rec.sport.baseball}, {sci.crypt, sci.electronics, sci.space}	6	5942

Table 3.3. Data Sets for Gene Expression Pairwise (Condition-Gene) Co-clustering

Name	Datasets	Data Structure	No. of clusters	No. of documents
BT1	ALL/AML	ALL, AML	2	72
BT2	Breast Cancer	Relapse, Non-relapse	2	97
BT3	Central Nervous	Class1, Class2	2	60
BT4	Colon Tumor	Positive, Negative	2	62
BT5	Lung Cancer	MPM, ADCA	2	181
BT6	Ovarian Cancer	Cancer, Normal	2	253
BT7	ALL/MLL/AML	ALL,MLL,AML	3	72

Table 3.4. Data Sets for Image High-Order (Color-Image-Texture) Co-clustering

Name	Datasets	No. of Modalities	No. of clusters	No. of documents
IT1	eggs,decoys	3	2	200
IT2	dawn,foliage	3	2	200
IT3	decoys,dawn	3	2	200
IT4	decoys,firearms,cards,buses	3	4	400
IT5	abstract,dawn,foliage,waves	3	4	400
IT6	eggs,decoys,dawn,foliage	3	4	400
IT7	eggs,decoys,buses,abstract,texture,dawn	3	6	600

3. http://datam.i2r.a-star.edu.sg/datasets/krbd/.

3.8. Results and Discussion

Performance analysis of RECCA and existing methods are Semisupervised Non-negative Matrix Factorization (SS-NMF) (Chen et al., 2010), Non-negative Matrix Factorization (NMF) (Xu et al., 2003), Combinatorial Markov Random Field (CMRF) (Bekkerman and Jeon, 2007), Semisupervised Combinatorial Markov Random Field (SS-CMRF) (Bekkerman and Sahami, 2006), Spectral Relational Clustering (SRC) (Long *et al.*, 2006) and Transductive Support Vector Machines (TSVM) (Joachims, 1999) is done in terms of accuracy and computation time.

Figure 3.1 uses the Text Pairwise (Document-Word) Co-clustering datasets depicted in Table 3.1.

Figure 3.2 uses the Gene Expression Pairwise (Condition-Gene) Co-clustering datasets depicted in Table 3.3.

Figure 3.3 uses the Text High-Order (Word-Document-Category) Co-clustering datasets depicted in Table 3.2.

Figure 3.4 uses the Image High-Order (Color-Image-Texture) Co-clustering datasets depicted in Table 3.4.

The experiments are performed on a Windows 8.1 machine with Intel Core i3 processors and 4 GB DDR III RAM. The experiments on algorithms are evaluated using MATLAB R2012a.

To measure the clustering results are measured by using the Accuracy Rate (AC), which compute how accurately a clustering algorithm assigns label y_i' to a data point with the ground truth y_i. The AC metric is defined as

$$AC = \frac{\sum_{i=1}^{n} \delta(y_i, y_i')}{n} \tag{3.26}$$

where n represents the total number of features in the experiment and δ is the delta function that equals one if $y_i' = y_i$; Elsewhere, it is zero.

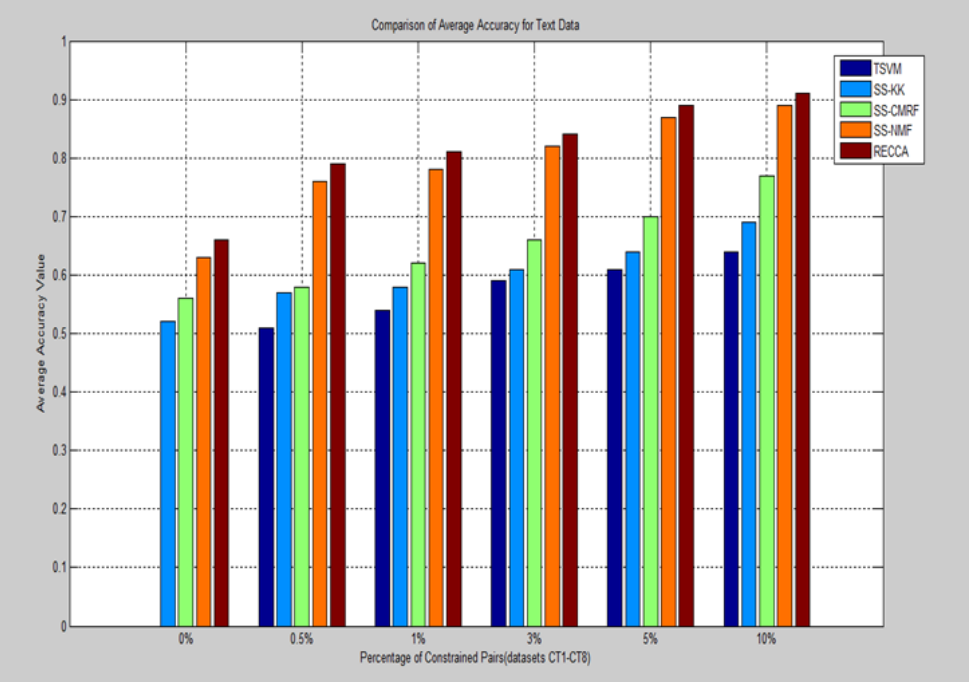

Figure 3.1. Comparison of Average Accuracy for Text Data

Figure 3.1 shows the performance evaluation of average accuracy for text data. It is evident that the proposed RECCA mechanism using Enhanced PCA outperforms other mechanisms in terms of document clustering performance with least prior knowledge. The performance values are depicted in Table 3.5.

Table 3.5. Comparison of Average Accuracy for Text Data

Algorithms / Percentage of Constrained Pairs	TSVM	SS-KK	SS-CMRF	SS-NMF	RECCA
0%	0	0.52	0.56	0.63	0.66
0.5%	0.51	0.57	0.58	0.76	0.79
1%	0.54	0.58	0.62	0.78	0.81
3%	0.59	0.61	0.66	0.82	0.84
5%	0.61	0.64	0.7	0.87	0.89
10%	0.64	0.69	0.77	0.89	0.91

Figure 3.2. Comparison of Average Accuracy for Gene Expression Data

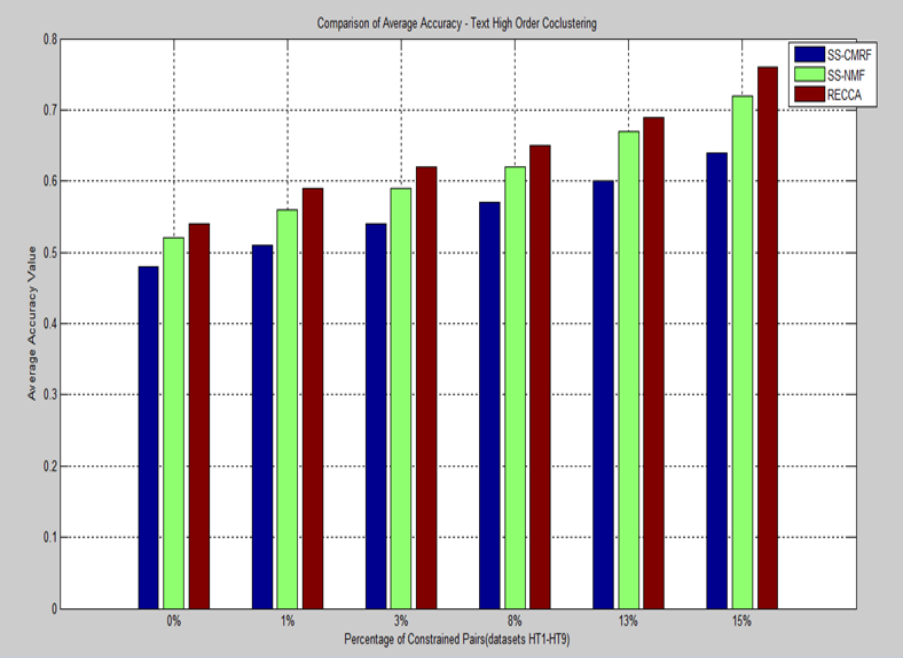

Figure 3.2 presents the performance evaluation of average accuracy for gene expression data. It is most visible that the proposed RECCA mechanism using Enhanced PCA outperforms other mechanisms in terms of increasing percentage of pairwise constraints for semisupervised condition co-clustering. The performance values are depicted in Table 3.6.

Table 3.6. Comparison of Average Accuracy for Gene Expression Data

Algorithms / Percentage of Constrained Pairs	TSVM	SS-KK	SS-CMRF	SS-NMF	RECCA
0%	0	0.54	0.66	0.59	0.61
0.5%	0.48	0.57	0.69	0.78	0.8
1%	0.54	0.59	0.73	0.8	0.83
3%	0.58	0.62	0.76	0.83	0.87
5%	0.62	0.67	0.79	0.86	0.89
10%	0.67	0.71	0.82	0.88	0.92

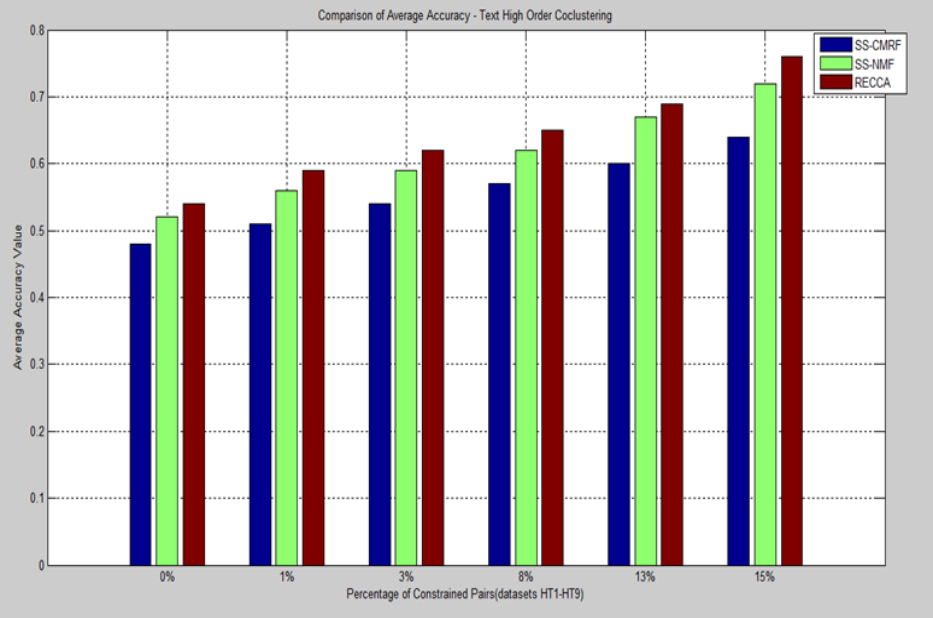

Figure 3.3. Comparison of Average Accuracy – Text High Order Co-clustering

Figure 3.3 presents the performance comparison of average accuracy for text high order co-clustering. It is most obvious that the proposed RECCA mechanism using Enhanced PCA outperforms other mechanisms. The performance values are depicted in Table 3.7.

Table 3.7. Comparison of Average Accuracy – Text High Order Co-clustering

Algorithms / Percentage of Constrained Pairs	SS-CMRF	SS-NMF	RECCA
0%	0.48	0.52	0.54
1%	0.51	0.56	0.59
3%	0.54	0.59	0.62
8%	0.57	0.62	0.65
13%	0.6	0.67	0.69
15%	0.64	0.72	0.76

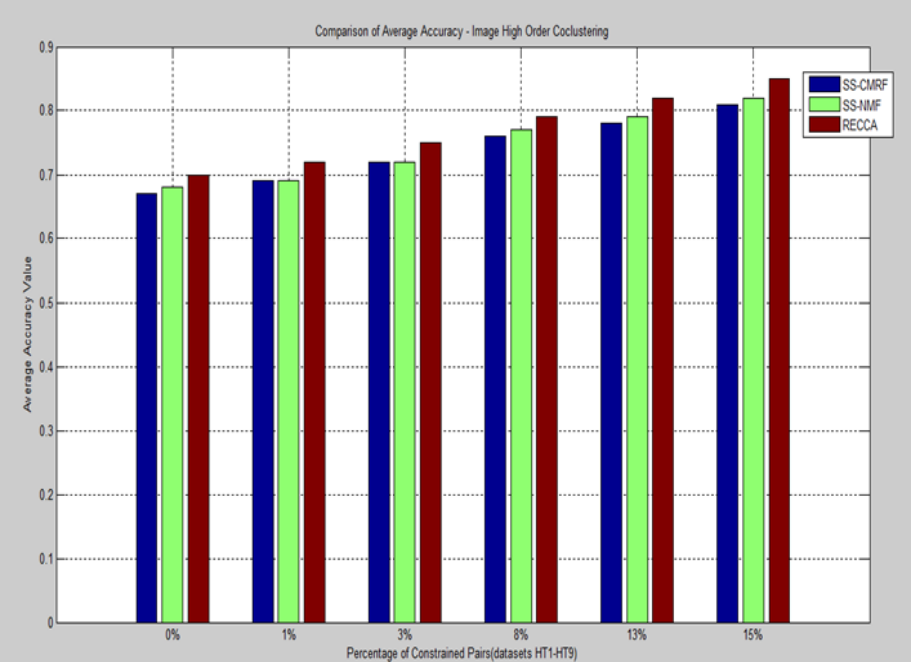

Figure 3.4. Comparison of Average Accuracy – Image High Order Co-clustering

Figure 3.4 presents the performance comparison of average accuracy for image high order co-clustering. It can be perceived that the proposed RECCA mechanism using Enhanced PCA outperforms other mechanisms. The performance values are depicted in Table 3.8.

Table 3.8. Comparison of Average Accuracy – Image High Order Co-clustering

Algorithms / Percentage of Constrained Pairs	SS-CMRF	SS-NMF	RECCA
0%	0.67	0.68	0.7
1%	0.69	0.69	0.72
3%	0.72	0.72	0.75
8%	0.76	0.77	0.79
13%	0.78	0.79	0.82
15%	0.81	0.82	0.85

Figure 3.5. Comparison of Computational Speed – In Time(Seconds) For Increasing N_c

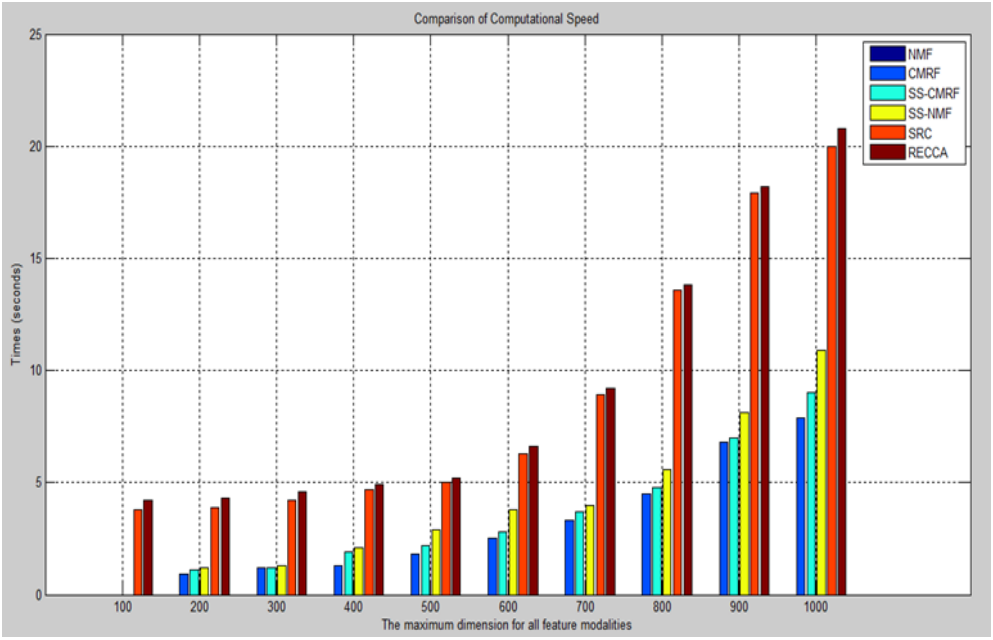

Table 3.9. Comparison of Computational Speed - In Time (Seconds) For Increasing N_c

Algorithms / Percentage of Constrained Pairs	NMF	SS-NMF	RECCA	CMRF	SS-CMRF	SRC
1000	0.05	0.23	0.21	3	6	10
1500	0.2	0.45	0.38	9	48	62
2000	0.1	0.56	0.52	34	69	89
2500	0.4	0.62	0.57	52	82	172
3000	0.52	0.84	0.74	92	107	352

Figure 3.6 presents the performance of computational time (the maximum feature dimension for all feature modalities - N_p) and the results proved that the proposed RECCA mechanism using Enhanced PCA approach delivers significant better performance over other methods. The performance values are depicted in Table 3.10.

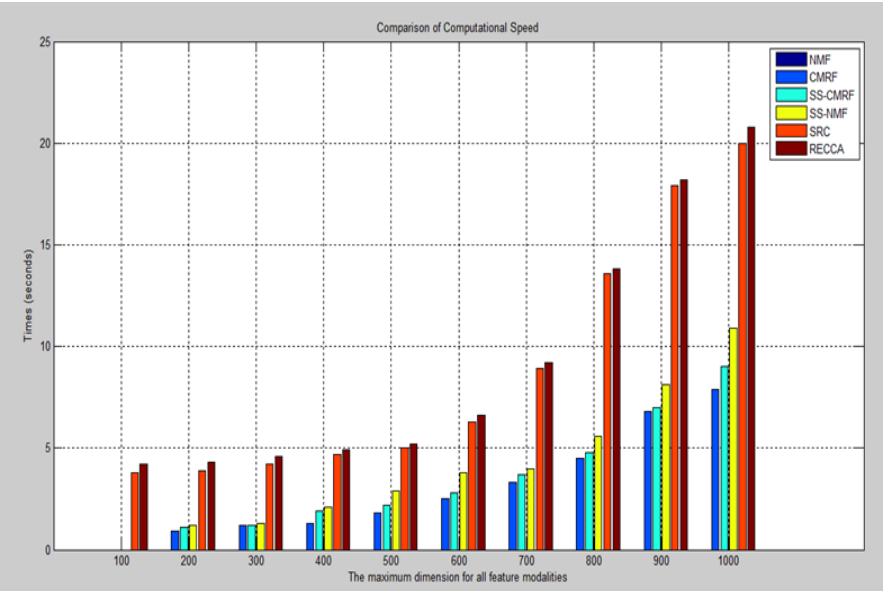

Figure 3.6. Comparison of Computational Speed - In Time (Seconds) For Increasing N_p

Table 3.10 Comparison of Computational Speed - In Time (Seconds) For Increasing N_p

Algorithms / Percentage of Constrained Pairs	NMF	CMRF	SS-CMRF	SS-NMF	SRC	RECCA
100	0	0	0	0	3.8	4.2
200	0.2	0.9	1.1	1.2	3.9	4.3
300	0.2	1.2	1.22	1.3	4.2	4.6
400	0.1	1.3	1.9	2.1	4.7	4.9
500	0.2	1.8	2.2	2.9	5	5.2
600	0.3	2.5	2.8	3.8	6.3	6.6
700	0.4	3.3	3.7	4	8.9	9.2
800	0.2	4.5	4.8	5.6	13.6	13.8
900	0.4	6.8	7	8.1	17.9	18.2
1000	0.2	7.9	9	10.9	20	20.8

3.9. Summary

In this chapter a mechanism is presented with improved preprocessing technique for clustering gene dataset samples. Initially the proposed work RECCA deals with the enhanced principal component analysis for preprocessing. The objective function for the co-clustering ensemble towards application to gene clustering is presented and it is also described. The objective function plays a major role which can perform co-clustering. Simulation results show that the proposed mechanism RECCA performs better in terms of accuracy and computation time. In this research work the RECCA is performed based on the co-clustering ensemble procedure. But in general if the dataset samples become high the computation complexity is required to complete clustering task. To reduce the computation complexity of RECCA algorithm in the next chapter ACO algorithms is introduced to group dataset samples.