International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878, Volume-8 Issue-4, November 2019

Learning an un-supervised – Clustering algorithm Monte Carlo over Consensus Clustering for Genomic Data for Tumor Identification

Tejal Upadhyay, Samir Patel

Abstract: ***Clustering*** *involves the grouping of similar objects into a set known as cluster. Objects in one cluster are likely to be different when compared to objects grouped under another cluster. Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. Subgroup classification is a basic task in high-throughput genomic data analysis, especially for gene expression and methylation data analysis. Mostly, unsupervised clustering methods are applied to predict new subgroups or test the consistency with known annotations. To get a stable classification of subgroups, consensus clustering is always performed. It clusters repeatedly with a randomly sampled subset of data and summarizes the robustness of the clustering. When faced with significant uncertainty in the process of making a forecast or estimation, the Monte Carlo Simulation might prove to be a better solution. Monte Carlo3C is a consensus clustering algorithm that uses a Monte Carlo simulation to eliminate overfitting and can reject the null hypothesis when only one cluster is there.*

Keywords: Consensus clustering, Monte Carlo Reference based Clustering, Genomic data, Supervised Learning Clustering

I. INTRODUCTION

1.1 Introduction

Consensus clustering: Clustering is an important property of unsupervised learning. Consensus clustering is a special type of clustering, when we have different types of clustering methods and it is desired to find a single clustering which has some sense than the existing clustering methods [1]. For the optimization problems, consensus clustering is known as median partition and has been shown to be NP-complete, even when the number of input clustering method is three. Consensus clustering for unsupervised learning is analogous to ensemble learning in supervised learning [2] [3]. There are some shortcomings for all existing clustering techniques. This may cause interpretation of results to become difficult, especially when there is no knowledge about the number of clusters. Consensus clustering provides a method that represents the consensus across multiple runs of a clustering algorithm, to determine the number of clusters in the data, and to assess the stability of the discovered clusters. The method can also be used to represent the consensus over multiple runs of a clustering algorithm with random restart (such as K-means, model-based Bayesian clustering, SOM, etc.), so as to account for its sensitivity to the initial conditions.[4]. Consensus Clustering [1] is a method that provides quantitative evidence for determining the number and membership of possible clusters within a dataset, such as microarray gene expression. This method has gained popularity in cancer genomics, where new molecular subclasses of disease have been discovered [3, 4].

1.2. Steps involve in Consensus Clustering

- Input: Microarrays
- Determine Clustering of specified cluster counts (k)
- The portion that two items occupied the same cluster – the same subsample are calculated and stored in symmetrical consensus matrix for each k.
- The consensus matrix is summarized in several graphical displays that enable a user to decide upon a reasonable cluster number and membership
- The Monti consensus clustering is a widely applied method to identify the number of clusters (K), however it has inherent bias towards greater values of K and yields high numbers of false positives.

When faced with significant uncertainty in the process of making a forecast or estimation, rather than just replacing the uncertain variable with a single average number, the Monte Carlo Simulation might prove to be a better solution [5].

II. CLUSTERING

2.1 Monte Carlo Reference Based Consensus Clustering:

M3C (Monte carlo Reference based consensus clustering uses a monte carlo

Manuscript published on November 30, 2019.
* Correspondence Author
Prof Tejal Upadhyay*, Assistant Professor, Department of Engineering, Gujarat University, Ahmadabad, India.
Dr Samir Patel, Assistant Professor, Department of Computer Science and Engineering, Pandit Deendayal Petrolium University, Raisan, Gandhinagar, Gujarat, India

simulation to generate all distributions of stability scores along the range of K which are used when deciding the optimal value. Using a reference, that maintains the correlation structures of the input feature, improves on the limitations of consensus clustering. M3C uses the Relative clustering Stability index (RCSI) and p values to decide on the value of K and reject the null hypothesis, k=1.

2.1.1 Operations of M3C:

A matrix of data frame of normalized continuous expression data (microarray) where columns equal samples and rows equal features.

Filtering is used for dimensionality reduction, filtered using variance for unsupervised and using p value for supervised.

Outliers are removed, with the PCA function which has a text label parameter

M3C only be used to cluster datasets with high numbers of samples (e.g. 75 and above). However, with sample sizes well over 1000, M3C will become highly time consuming so there is a upper limit. (75-1000 samples).

M3C also accepts clinical or biological data for a statistical analysis with the discovered classes.

Again the 'ID' column must exist in the annotation data frame. If the data is continuous or categorical, a Kruskal-Wallis or chi-squared test are performed, respectively.

For these two tests, a 'variable' parameter must be given which is a string that defines the dependent variable in the users annotation data frame. If the data is survival data for cancer, the annotation data frame must have an 'ID' column first, followed by a 'Time' and 'Death' column (where 0 is no death and 1 is death or time until last visit).

2.2 Workflow I: TCGA Glioblastoma Dataset

The glioblastoma (GBM) cancer dataset is used and randomly 50 samples are taken

2.3 Exploratory data analysis

Before proceeding to run M3C function, we should remove the extreme outliers. After that we are using the following methods PAM (Partition Around Medoids), K-means, and HC (Hierarchical clustering).

2.3.1 K-Means

Trying to separate samples in n groups of equal variance, minimizing a criterion known as the inertia or within-cluster sum-of-squares. The number of clusters to be specified.

The k-means algorithm divides a set of N samples X into K disjoint clusters C, each described by the mean μj of the samples in the cluster.

2.3.2 PAM (Partition Around Medoids)

Similar to K Means but *k*-medoids chooses data points as centers (medoids) and can be used with arbitrary distances

It is a classical partitioning technique of clustering, which clusters the data set of *n* objects into *k* clusters, with the number *k* of clusters assumed known a priori

A medoid can be defined as the object of a cluster whose average dissimilarity to all the objects in the cluster is minimal, that is, it is a most centrally located point in the cluster.

2.3.3 HC (Hierarchical clustering)

It is a general family of clustering algorithms that build nested clusters by merging or splitting them successively. Represented as a tree (or dendrogram)

The root of the tree is the unique cluster that gathers all the samples, the leaves being the clusters with only one sample

For these methods the following assumption is considered.

Clusters are approximately spherical - not severely elongated in one direction (anisotropic) or non-linear [6]

Principal Component Analysis (PCA) is used to explain the variance-covariance structure of a set of variables through linear combinations. It is often used as a dimensionality-reduction technique the following figure shows that Clusters are approximately equal in variance.

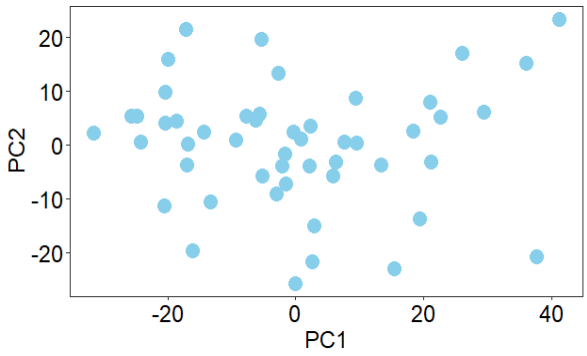

Figure 1Principal Component Analysis

III. RESULTS

3.1 Running M3C

In our example, we have used the Monte Carlo reference based clustering algorithm with the Monte Carto Simulation with the following parameter is used.

M3C is a consensus clustering algorithm that improves performance by eliminating overestimation of K and can test the null hypothesis K=1 [7].

M3C calculates the consensus rate, a measure of stability of the co-clustering of samples, which is quantified for each K using the PAC score -Generation of reference PAC distribution using a multi-core Monte Carlo simulation -Reference generation preserves feature-feature correlation structure of data -Using the reference distributions the Relative Cluster Stability Index (RCSI) and empirical p values are used to select K and reject the null, K=1. -Extrapolated p values are calculate by fitting a beta distribution -A second method is included for faster results that uses a penalty term instead of a Monte Carlo simulation to deal with bias towards higher values of K -Automatic re ordering of expression matrix and annotation data to help user do their analysis faster -Automatic analysis of clinical or biological data using survival analysis, chi-squared, or Kruskal-Wallis tests -User friendly PCA, tSNE, and UMAP functions that interface with the results -All plotting code using ggplot2 for publication quality outputs.

We present Spectrum, a fast spectral clustering method for single and multi-omic expression data. Spectrum is flexible and performs well on single-cell RNA-seq data.

The method uses a new density-aware kernel that adapts to data scale and density. It uses a tensor product graph data integration and diffusion technique to reveal underlying structures and reduce noise [8].

res <- M3C(mydata, cores=1, seed = 123, des = desx, removeplots = TRUE, analysistype = 'chi', doanalysis = TRUE, variable = 'class')

Input: A matrix or data frame of normalized continuous expression data (e.g. microarray, RNA-seq, methylation arrays, protein arrays) in which columns are samples and rows are features.
Mydata: Data frame or matrix: Contains the data.
Cores: Numerical value: how many cores to split the Monte Carlo simulation over
Seed: Numerical value: fixes the seed if you want to repeat results, set the seed to 123 for example here
Des: Data frame: contains annotation data for the input data for automatic reordering
Removeplots: Logical flag: whether to remove all plots
Analysistype: Character string: refers to which kind of statistical analysis to do on the data, survival, Kruskal-Wallis (kw), or chi-squared (chi)
Doanalysis: Logical flag: whether to analyse the clinical variable supplied (univariate only)
Variable: Character string: if not doing survival what is the dependent variable (column name) called in the data frame.
P-value of a cluster is a value between 0 and 1, which indicates how strong the cluster is supported by data. M3C uses by default PAM with Euclidean distance in the consensus clustering loop because we have found this runs fast with good results. The Monte Carlo simulations maintains the feature correlation structure of the input data. Then the null distribution is used to compare the reference scores with the real scores and an empirical p value is calculated for every value of K to test the null hypothesis K=1. We derive the Relative Cluster Stability Index (RCSI) as a metric for selecting K, which is based on a comparison against the reference mean. A faster alternative is included that includes a penalty term to prevent overfitting, called the Penalised Cluster Stability Index (PCSI).

```
> res$scores
   K PAC_REAL PAC_REF       RCSI MONTECARLO_P    BETA_P  P_SCORE
1  2 0.6734694 0.5773796 -0.15394262   0.66336634 0.69721054 0.1566361
2  3 0.4473469 0.5410857  0.19024326   0.18811881 0.19923640 0.7006313
3  4 0.3404082 0.4711755  0.32508528   0.03960396 0.03313314 1.4797374
4  5 0.3200000 0.4018041  0.22764361   0.08910891 0.07848793 1.1051971
5  6 0.3102041 0.3474204  0.11330519   0.23762376 0.22889997 0.6403543
6  7 0.2840816 0.3031673  0.06502332   0.31683168 0.33674564 0.4726980
7  8 0.2653061 0.2700408  0.01768878   0.45544554 0.46514316 0.3324134
8  9 0.2465306 0.2413469 -0.02125070   0.56435644 0.56967334 0.2443741
9 10 0.2130612 0.2190531  0.02773443   0.44554455 0.44629390 0.3503790
```

The above results shows that maximum value of RSCI is 0.33 and the corresponding K value is 4, the Monte Carlo p value supports 0.039. p value shows that This means the null hypothesis that K = 1 can be rejected for this dataset because we have achieved significance (alpha=0.05) versus a dataset with no clusters. For p values that extend beyond the lower limits imposed by the Monte Carlo simulation, M3C estimates parameters from the simulation to generate a beta distribution. The BETA_P in this case study is 0.033.
Also important is the relationship between the clinical variables and the discovered clusters. In this data we want to compare with a categorical variable so perform a chi-squared test. We are reassured to see below K=4 is highly significant (p=5.5x10-14), however, K=5 is slightly more so. It is important to bear in mind the clinical or biological relationships as well as the structural nature of the data when deciding K.
This is a CDF (Cumulative distribution function) plot of the consensus matrices for our test data. We are looking for the value of K with the flattest curve and this can be quantified using the PAC metric. In the CDF and following PAC plot we can see the overfitting effect of consensus clustering whereas K increases so does the apparent stability of the results for any given dataset, this we correct for by using a reference [9].

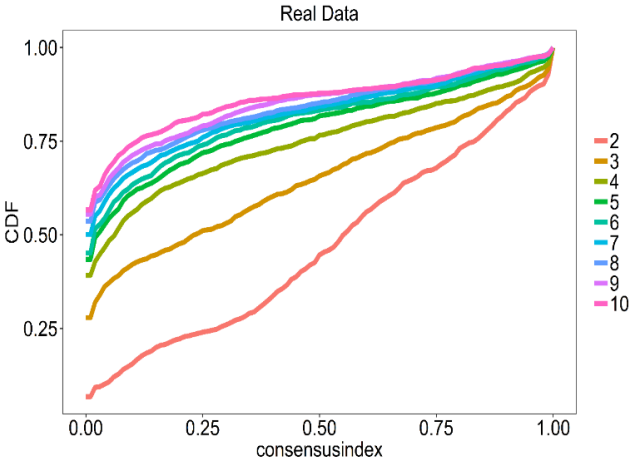

Figure 2Cumulative Distribution Function for real data

The following diagram is the PAC (The proportion of ambiguous clustering) score, we can see an elbow at K = 4 which is suggestive this is the best K. However, the overfitting problem of the PAC score can be seen here as it tends towards lower values as K increases, thus making selecting K without taking this into account challenging to say the least. Furthermore, the PAC score cannot reject the null hypothesis K=1, for that we have introduced p values into the consensus clustering methodology.

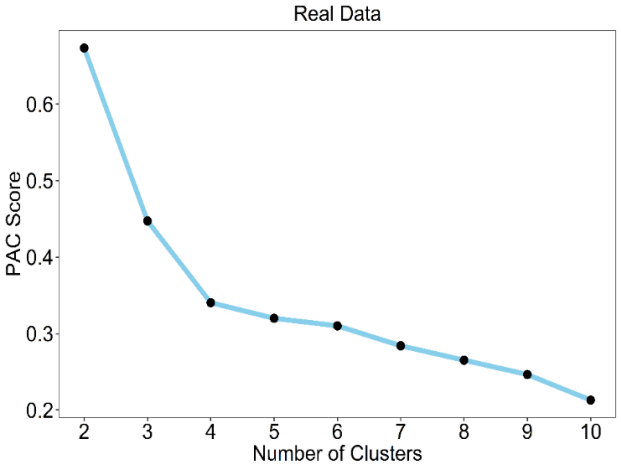

Figure 3 PAC score of real data

We then derive the Relative Cluster Stability Index (RCSI) which takes into account the reference PAC scores using the reference mean. This metric is better than the PAC score for deciding class number, where the maximum value corresponds to the optimal K. In this example the RCSI has an optima at K=4.

We recommend the RCSI be used to select K, and the p values to reject the null in most cases. This metric is better than the PAC score for deciding class number, where the maximum value corresponds to the optimal K. In this example the RCSI has an optima at K=4. We recommend the RCSI be used to select K, and the p values to reject the null in most cases.

Figure 4 RCSI(Relative Cluster Stability Index)

Finally, we calculate a p value from the distribution, here we display the p values from the beta distribution. If none of the p values reach significance over a reasonable range of K (e.g. 10), then we accept the null hypothesis. In the GBM dataset, we can see K = 4 reaches signfiicance with an alpha of 0.05 (red dotted line), therefore we can reject the null hypothesis K=1 for the GBM dataset.

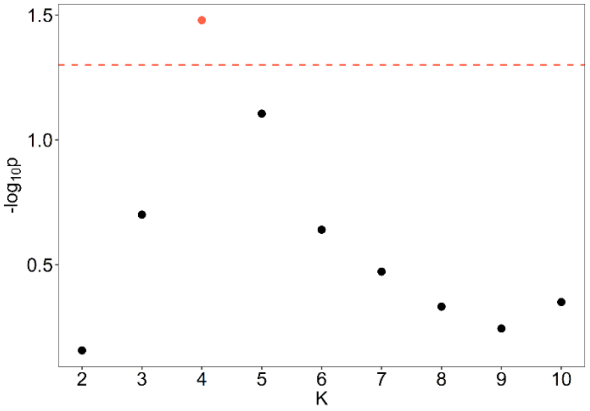

Figure 5 Empirical p Values

Now we are pretty convinced there are 4 clusters within this dataset which are not likely simply to have occurred by chance alone.

A further analysis that M3C conducts (if the dend flag is set to TRUE) is to make a dendrogram for the optimal K using the consensus cluster medoids, then sigclust is run to calculate p values for each branchpoint. This allows quantification of the structural relationships between consensus clusters. In this case CC3 and CC4 have a closer relationship, whilst the other clusters are quite far apart.

3.2 M3C outputs

The following lines extract the ordered (according in clustering results) expression data and the ordered annotation data from the results object after running M3C for a 4 cluster solution. We then take a brief glance at the annotation object M3C outputs, a consensus cluster column has been added by M3C [10].

```
# get the data out o  the results list (by using $ - dollar sign) for K=4
data <- res$realdataresults[[4]]$ordered_data # this is the data
annon <- res$realdataresults[[4]]$ordered_annotation # this is the annotation
ccmatrix <- res$realdataresults[[4]]$consensus_matrix # this is the consensus matrix
```

Next, we scale the data here row wise according to z-score, prior to some light data compression for visualisation purposes in the heatmap.

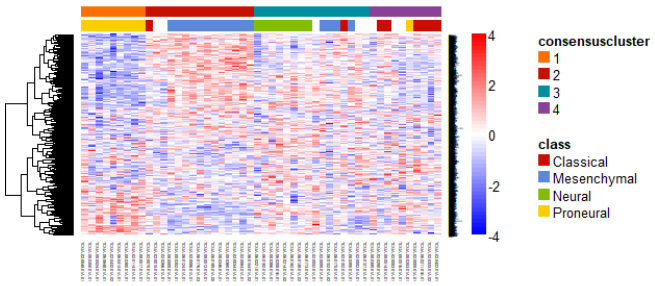

Figure 6 Aheatmap of GBM consensus clusters with tumour classification

Another plot shows the consensus matrix for our optimal clustering solution (in this case, K = 4). We can see in this heatmap below of the consensus matrix the clusters do indeed look quite clear supporting our view that there is 4 clusters.

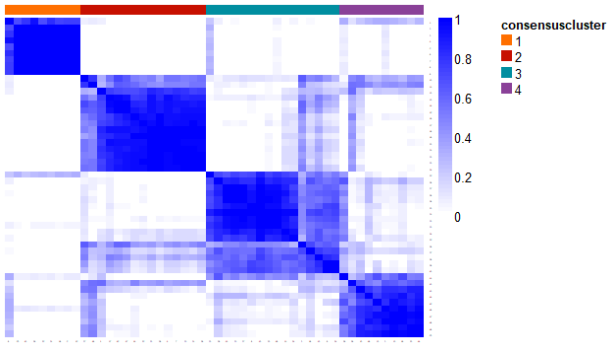

Figure 7 Aheatmap of GBM consensus matrix

Viewing consensus matrices manually should not be used to decide K, better to use the RCSI or p values that M3C provides to quantify the mixing proportions in the consensus matrix versus the K=1 null model[11][12].

We also caution against the use of cluster validity metrics on the consensus matrices without first testing their behaviour in details on simulated positive control datasets. Our data indicates that these metrics may have substantial bias in them as well due to the underlying method.

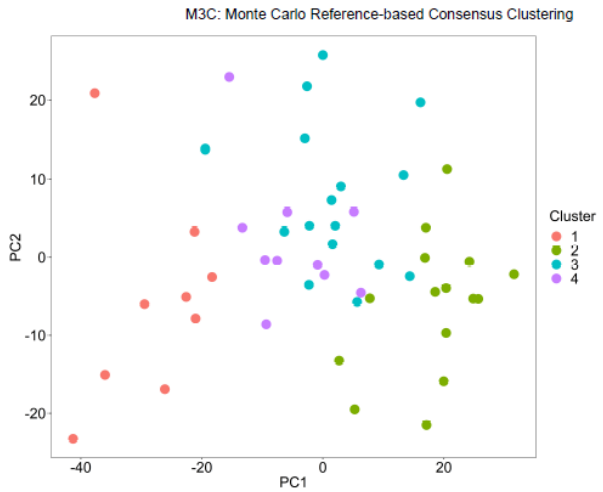

Figure 8 PCA of a 4 cluster test dataset

A last analysis we recommend to do is to examine how the clusters are arranged in principal component and t-SNE latent variable space.

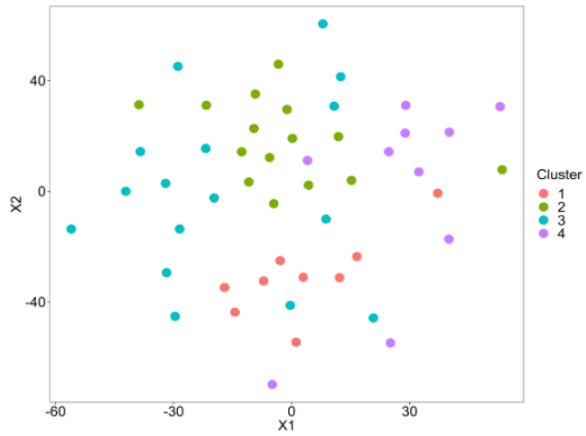

Figure 9 t-SNE of 4 cluster simulated dataset

IV. CONCLUSION

The consensus clustering is the aggregation of clustering, where a number of different (input) clustering have been obtained for a particular dataset and it is desired to find a single (consensus) clustering. When we need to forecast and facing with significant uncertainty in the process, Monte Carlo Simulation may give better result. So we have used M3C base clustering and get some results. Before applying M3C, we have applied some pre-processing techniques like K-means, PAM and HC. M3C is a consensus clustering algorithm that improves performance by eliminating overestimation of K and can test the null hypothesis K=1.

The CDF (Cumulative distribution function) plot of the consensus matrices for our test data shows that the best k value for this dataset is 4. Another diagram shows that the PAC score cannot reject the null hypothesis K=1, for that we have introduced p values into the consensus clustering methodology.

Finally, we calculate a p value from the distribution and displayed over a reasonable range of K values. If none of the p values reach significance over a reasonable range of K (e.g. 10), then we accept the null hypothesis. In the GBM dataset, we can see K = 4 reaches significance with an alpha of 0.05 (red dotted line), therefore we can reject the null hypothesis K=1 for the GBM dataset.

The next diagram shows the Tumor Classification of four different classes with four clusters. Final Analysis is shows how the clusters are arranged in Principal Component Analysis and t-SNE variables space.

REFERENCES

1. Alexander Strehl and J. Ghosh, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, *Journal on Machine Learning Research (JMLR) 2002, 3, 583-617*.
2. Punera, Kunal, and Joydeep Ghosh. "Consensus-based ensembles of soft clusterings." *Applied Artificial Intelligence* 22.7-8 (2008): 780-810..
3. Gionis, Aristides, Heikki Mannila, and Panayiotis Tsaparas. "Clustering aggregation." ACM Transactions on Knowledge Discovery from Data (TKDD) 1.1 (2007): 4.
4. Wang, Hongjun, Hanhuai Shan, and Arindam Banerjee. "Bayesian cluster ensembles." Statistical Analysis and Data Mining: The ASA Data Science Journal 4.1 (2011): 54-70.
5. Nguyen, Nam, and Rich Caruana. "Consensus clusterings." Seventh IEEE International Conference on Data Mining (ICDM 2007). IEEE, 2007
6. Topchy, Alexander, Anil K. Jain, and William Punch. "Clustering ensembles: Models of consensus and weak partitions." IEEE transactions on pattern analysis and machine intelligence 27.12 (2005): 1866-1881
7. John, Christopher Robert, et al. "M3C: A Monte Carlo reference-based consensus clustering algorithm." bioRxiv (2018): 377002
8. John, Christopher R., et al. "Spectrum: Fast density-aware spectral clustering for single and multi-omic data." BioRxiv (2019): 636639.
9. Monti, Stefano, et al. "Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data." Machine learning 52.1-2 (2003): 91-118.
10. Kvålseth, Tarald O. "Coefficient of variation: the second-order alternative." Journal of Applied Statistics 44.3 (2017): 402-415.
11. Şenbabaoğlu, Yasin, George Michailidis, and Jun Z. Li. "Critical limitations of consensus clustering in class discovery." Scientific reports 4 (2014): 6207
12. Tibshirani, Robert, Guenther Walther, and Trevor Hastie. "Estimating the number of clusters in a data set via the gap statistic." Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63.2 (2001): 411-423

AUTHORS PROFILE

Prof Tejal Upadhyay has completed Bachelor of Engineering from Gujarat University, Ahmedabad, Gujarat and Master of Engineering from Dharamsinh Desai University and at present she is pursuing PhD. Her Research area is working with Genome data using data mining techniques. She has a more than 22 years' experience of teaching to under graduate and post graduate students of Compute Engineering / Information Technology. At present she is working as an Assistant Professor at Nirma Univesity since year 2000. She has a professional memberships like ISTE (Indian Society for Technical Education), CSI (Computer Society of India) etc. She is also Student Branch Counselor for CSI student Branch at Nirma University since 2005 to till date. Under CSI she has organized many technical events like coding competition, workshops, expert talks, other managerial competitions and many more.

Dr Samir Patel has total of 20 years of teaching experience. currently he is working as an Assistant Professor at Pandit Deendayal Petroleum University, SOT, CSE Department. Earlier, he has worked at the head positions at various places like Chimanbhai Patel Institute, P.D. Pandya Institute at AES Institute of Computer Studies.

He has started working as Sr. Lecturer and then was promoted as Assistant Professor subsequently he had joined Nirma University as Associate Professor and then was promoted as Sr. Associate Professor. He has obtained his Doctoral degree under the able guidance of Dr. S. N. Pradhan (Retd. Sr. Scientist from PRL and Head M. Tech. CSE – Nirma University). He got the opportunity to work in the capacity of principal for 4 and half years at Grow More Faculty of Engineering-Himatnagar. His area of interest includes Data computing, Multimedia data processing, Data mining, Parallel computing, Image Processing, etc. He had published 24 research papers in International and National Journals/conferences. He has visited France under the Travel Grant of AICTE for his research work during his Ph.D. work. He authored one book of "An Integration of Image Processing and Data Mining to Perform Digital Watermarking" in the year 2014. He had delivered several expert lectures in the colleges of repute. He has organized many STTP/FDPs/Seminars in different areas of interest. He is a life member of professional bodies like Computer Society of India and ISTE. He is awarded by GTU for his Pedagogical Innovation in February 2014.

Retrieval Number: D7370118419/2019©BEIESP
DOI:10.35940/ijrte.D7370.118419
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)