

**A Novel Approach in Clustering of Data Set
using Self-Organizing Map**

A thesis

submitted in partial fulfillment of the requirement for the Degree of

Master of Computer Science and Engineering

of

Jadavpur University

By

Arindam Roy

Registration No.: 128995 of 2014-15

Examination Roll No.: M4CSE1608

Under the Guidance of

Prof. Nirmalya Chowdhury

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2016

**A Novel Approach in Clustering of Data Set
using Self-Organizing Map**

A thesis

submitted in partial fulfillment of the requirement for the Degree of

Master of Computer Science and Engineering

of

Jadavpur University

By

Arindam Roy

Registration No.: 128995 of 2014-15

Examination Roll No.: M4CSE1608

Under the Guidance of

Prof. Nirmalya Chowdhury

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2016

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Recommendation

This is to certify that the dissertation entitled “A Novel Approach in Clustering of Data Set using Self-Organizing Map” has been carried out by Arindam Roy (University Registration No.: 128995 of 2014-15, Examination Roll No.: M4CSE1608) under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Master of Computer Science and Engineering. The research results presented in the thesis have not been included in any other paper submitted for the award of any degree in any other University or Institute.

.....
Prof. Nirmalya Chowdhury (Thesis Supervisor)
Department of Computer Science and Engineering
Jadavpur University, Kolkata-32

Countersigned

.....
Prof. Debesh Kumar Das
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-32.

.....
Prof. Sivaji Bandyopadhyay
Dean, Faculty of Engineering and Technology,
Jadavpur University, Kolkata-32.

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Certificate of Approval*

This is to certify that the thesis entitled “A Novel Approach in Clustering of Data Set using Self-Organizing Map” is a bona-fide record of work carried out by Arindam Roy in partial fulfillment of the requirements for the award of the degree of Master of Computer Science and Engineering in the Department of Computer Science and Engineering, Jadavpur University during the period of June 2015 to May 2016. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

.....

Signature of Examiner 1

Date:

.....

Signature of Examiner 2

Date:

*Only in case the thesis is approved

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled “A Novel Approach in Clustering of Data Set using Self-Organizing Map” contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Computer Science & Engineering.

All information have been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Arindam Roy

Registration No: 128995 of 2014-15

Exam Roll No.: M4CSE1608

Thesis Title: A Novel Approach in Clustering of Data Set using Self-Organizing Map

.....
Signature with Date

Acknowledgement

I express my sincere gratitude to **Prof. Nirmalya Chowdhury**; my guide for his affectionate and valuable guidance without whose help the present work could not have been a successful one. I am also indebted to him as a Professor who introduced me to the world of Data Mining and Pattern Recognition.

Also I thank **Prof. Debesh Kumar Das**, Head of the Department of Computer Science and Engineering, for his assistance in allowing me to work in the departmental laboratory without which my work would have been incomplete.

I would also like to convey my sincere gratitude to all my respected teachers and faculty members in this department for their invaluable suggestions and kind cooperation.

I express my thanks to all friends of my class.

Last but not the least, the encouragement given by my mother **Mrs. Hasi Roy**, my father **Mr. Chandan Roy** who have always been a constant source of inspiration and whose encouragement is beyond linguistic expression for me.

Arindam Roy
Jadavpur University, Kolkata

.....
Arindam Roy

Registration No: 128995 of 2014-15

Exam Roll No.: M4CSE1608

Department of Computer Science & Engineering
Jadavpur University

Table of Contents

	Page No.
Chapter 1: INTRODUCTION	1-28
1.1 Introduction to Pattern Recognition	
1.1.1 What is Pattern Recognition?	
1.1.2 Different Approaches in Pattern Recognition	
1.1.3 Pattern Recognition Methods	
1.1.3.1 Supervised Method	
1.1.3.2 Semi-supervised Method	
1.1.3.3 Unsupervised Method	
1.2 Introduction to Soft Computing	
1.2.1 Artificial Neural Network	
1.2.2 Fuzzy Logic	
1.2.3 Genetic Algorithm	
1.3 Application of Pattern Recognition Techniques	
 Chapter 2: CLUSTERING TECHNIQUES	 29-53
2.1 Introduction	
2.2 Clustering Techniques	
2.2.1 Hierarchical Clustering	
2.2.1.1 Agglomerative Nesting	
2.2.1.1.1 Single Linkage	
2.2.1.1.2 Complete Linkage	
2.2.1.1.3 Average Linkage	
2.2.1.2 Divisive Analysis	
2.2.1.3 BIRCH	
2.2.1.4 CURE	
2.2.1.5 ROCK	
2.2.1.6 CHAMELEON	
2.2.2 Partitional Clustering	
2.2.2.1 K-means	

- 2.2.2.2 ISODATA
- 2.2.2.3 Partitioning Around Medoids (PAM)
- 2.2.2.4 CLARANS
- 2.2.2.5 K-mode
- 2.2.3 Distance Based Clustering
 - 2.2.3.1 Nearest Neighbor Clustering
- 2.2.4 Fuzzy Clustering
 - 2.2.4.1 Fuzzy c-means
 - 2.2.4.2 Gustafson Kessel
- 2.2.5 Evolutionary Clustering
 - 2.2.5.1 GA-Based Clustering
 - 2.2.5.2 Variable Length GA
- 2.2.6 Model-Based Clustering
 - 2.2.6.1 Expectation-Maximization
 - 2.2.6.2 Artificial Neural Network Based Clustering
- 2.2.7 Graph-Based Clustering
 - 2.2.7.1 Minimum Spanning Tree Based Clustering
- 2.2.8 Density Based Clustering
 - 2.2.8.1 DBSCAN
 - 2.2.8.2 OPTICS
 - 2.2.8.3 DENCLUE
- 2.2.9 Grid Based Clustering
 - 2.2.9.1 STING
 - 2.2.9.2 Wave Cluster
- 2.2.10 Subspace Clustering
 - 2.2.10.1 CLICK
 - 2.2.10.2 CLIQUE

Chapter 3: KOHONEN SELF-ORGANIZING MAP

54-67

- 3.1 Introduction
- 3.2 Discussion of Self-Organizing Map Algorithm

Chapter 4: LITERATURE REVIEW	68-76
4.1 Self-Organizing Map in Clustering	
4.2 MST Based Clustering	
Chapter 5: PROPOSED METHOD OF CLUSTERING	77-80
5.1 Statement of the Problem	
5.2 Detailed Description of the Proposed Method	
5.3 Proposed Method in the form of an Algorithm	
Chapter 6: IMPLEMENTATION OF PROPOSED METHOD IN IMAGE SEGMENTATION	81-86
6.1 What is Image Segmentation?	
6.2 Traditional Image Segmentation Techniques	
6.2.1 Color based segmentation	
6.2.2 Histogram Based Segmentation	
6.2.3 Region Based Segmentation	
6.3 Segmentation of an RGB Color Image using Proposed Method	
6.3.1 Brief Discussion of Proposed Segmentation Procedure	
6.3.2 Flow chart of Proposed Segmentation Method	
Chapter 7: EXPERIMENTAL RESULTS	87-99
7.1 Software Requirements	
7.2 Description of the Experimental Results	
7.2.1 Results on Synthetic Data Sets	
7.2.2 Results on Real Life Data Sets	
7.3 Comparison of Results with other Clustering Methods	
7.4 Results on Image Data Sets for Segmentation	
Chapter 8: CONCLUSION & SCOPE FOR FURTHER RESEARCH	100-101
8.1 Conclusion	
8.2 Scope for Further Research	

REFERENCES

LIST OF FIGURES

Figure No.	Figure Name	Page No.
FIG-1	Two Modes of Pattern Recognition System	3
FIG-2	Steps in a typical PR process	3
FIG-3	A Pattern Classification System	4
FIG-4	Linear Decision Function	6
FIG-5	Flow Diagram of the Supervised Learning	11
FIG-6	Semi supervised Method	12
FIG-7	SCHEMETIC DIAGRAM OF A BIOLOGICAL NEURON	17
FIG-8	Artificial Neural Network	18
FIG-9	SCHEMETIC STRUCTURE OF AN ANN	19
FIG-10	A GEOMETRIC INTERPRETATION OF THE ROLE OF HIDDEN LAYER	20
FIG-11	TAXONOMY OF NEURAL NETS	21
FIG-12	A Comparison of Membership Function Crisp vs. Fuzzy Set	23
FIG-13	A Fuzzy Logic System	24
FIG-14	Different stages of Clustering Technique	30
FIG-15	Different type of Clustering Technique	31
FIG-16	Graphical Example of Single Linkage Method	33
FIG-17	Graphical Example of Complete Linkage Method	34
FIG-18	Graphical Example of Average Linkage Method	35
FIG-19	CF Tree	36
FIG-20	Flow of CURE Procedure	37
FIG-21	CURE Approach	38
FIG-22	CHAMELEON	39
FIG-23	K-Means Clustering	40
FIG-24	Hard vs. Fuzzy Clustering	44
FIG-25	Graph and corresponding MST	49
FIG-26	MST Based Clustering	49

FIG-27	Schematic diagram of Kohonen Self Organizing Map	56
FIG-28	Rectangular and Hexagonal Neighborhood	58
FIG-29	SOM Grid Structure	58
FIG-30	Winner Neuron and their neighbour nodes in SOM	59
FIG-31	Outline of the Proposed Method of Image Segmentation	86

Chapter 1: INTRODUCTION

1.1 Introduction to Pattern Recognition

Pattern:-

A pattern is defined by the common denominator among the multiple instances of an entity [93]. Based on the common features among those instances of an entity a pattern can be extracted. For example based on common features in all the finger print images we can extract a pattern of finger print. A pattern could be a finger print image, handwritten word, human face, speech signal, DNA sequence.

A pattern class is a family of patterns that share some common properties. Pattern classes are denoted $\omega_1, \omega_2, \dots, \omega_n$ where w is the number of classes [94]. To extract a pattern from data there are several stages described below.

- 1) Data Cleaning: - This stage is used to remove noise and inconsistent data.
- 2) Data Integration: - In this stage multiple data sources may be combined.
- 3) Data Selection: - Data relevant to the analysis task are retrieved from the data base.
- 4) Data Transformation: - Data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance.
- 5) Data Mining: - In this process some intelligent methods like clustering, classification are applied in order to extract the data pattern.
- 6) Pattern Extraction and Evaluation:- In this stage patterns are extracted and evaluated to get knowledge.

1.1.1 What is Pattern Recognition?

Pattern recognition is the scientific discipline whose goal is the classification of objects into a number of categories or classes. Depending on the application these objects can be images or signal waveforms or any type of measurement that needs to be classified [95]. Pattern Recognition is a branch of machine learning that focuses on the recognition of patterns and regularities in data. There are two approaches in pattern recognition techniques defined below.

1) Statistical Pattern Recognition

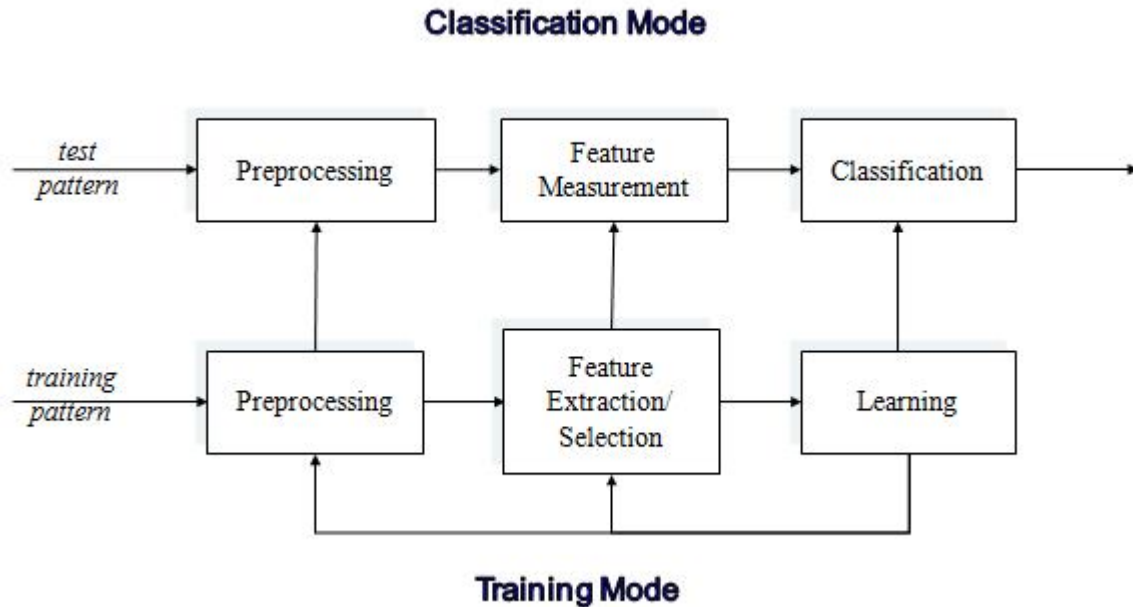
The data is reduced to vectors of numbers and statistical techniques are used for the tasks to be performed. This type of pattern recognition technique is based on underlying statistical model of patterns and pattern classes.

2) Structural Pattern Recognition

Here pattern classes are represented by means of formal structures as grammars, automata, strings, etc. The data is converted to a discrete structure (such as a grammar or a graph) and the techniques are related to computer science subjects (such as parsing and graph matching).

Pattern Representation:-

- ▶ A pattern is represented by a set of d features, or attributes, viewed as a d -dimensional vector $x=(x_1, x_2, \dots, x_d)^T$, it is called a pattern vector or feature vector.

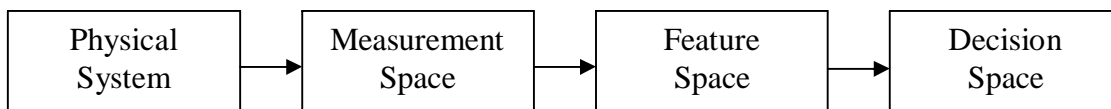
Two Modes of Pattern Recognition System:-**FIG-1: Two Modes of Pattern Recognition System**

The problem of pattern recognition can be divided into two parts:

- I. It is concerned with the study of recognition mechanism of patterns by human and other living organism. This part is related to the disciplines like physiology, psychology, biology, etc.
- II. It deals with the development of theory and techniques for designing a device which can perform these recognition tasks automatically. This area is related to engineering, computer and information sciences. In this curriculum, we shall be dealing with the second part i.e. the problems of automatic machine recognition of patterns.

Operating Stages in a Pattern Recognition System:-

The operating stages those are necessary to develop a pattern classifier are stated below.

**FIG-2: Steps in a typical PR process**

Physical System:-

This stage basically deals with the collection of data that may be characterized by some of its physical properties, which is able to measure on appropriate variables.

Measurement Space:-

The system from which given patterns arise is characterized completely only by its physical embodiment. We characterized that embodiment numerically by some set of measurements that form the measurement space. A sample of pattern is represented by specific values of all the measurements, corresponding to a point in the measurement space.

Feature Space:-

The pattern classification algorithms should be applied in a feature space which is finite-dimensional and contains sufficient information to satisfactorily perform the recognition. A point in the measurement space is transformed by the intermediate processing into a point $x = (x_1, x_2, \dots, x_n)$ in feature space, where each x_i ($i \in 1$ to n) is called a feature. By the process of feature selection or preprocessing a smaller set of features are extracted from feature space, which forms the reduced feature space.

Decision Space:-

In the decision space we must develop, on the basis of finite set of labeled samples, a decision rule with which we can classify a point in the feature space corresponding to an unlabeled sample, assign a pattern to a specific pattern class, so this process is called pattern classification.

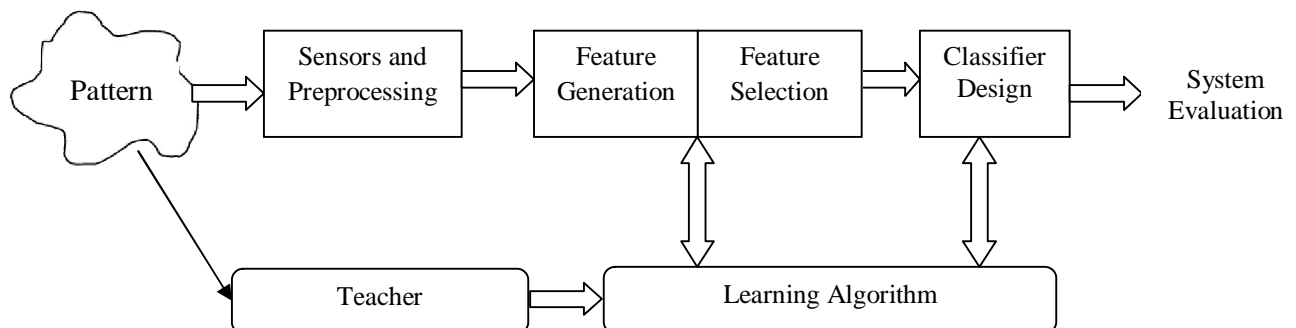
Components of Pattern Recognition:-

FIG-3: A Pattern Classification System

Above figure shows the various stages followed for the design of a pattern classification system. As is apparent from the feedback arrows, these stages are not independent. On the contrary, they are interrelated and, depending on the results, one may go back to redesign earlier stages in order to improve the overall performance.

▶ Sensors and preprocessing:-

The input to a pattern recognition system is often some kind of a transducer, such as a camera or a microphone array. The difficulty of the problem may well depend on the characteristics and limitations of the transducer- its bandwidth, resolution, sensitivity, distortion, signal-to-noise ratio, latency, etc. So there is a high probability that noise may be present in input data. To remove noise from data preprocessing is required.

▶ Feature Generation and Feature Selection:-

Select variables from the measured set those are appropriate for the task. These new variables may be obtained by a linear or nonlinear transformation of the original set (feature extraction). To some extent, the partitioning of the data processing into separate feature extraction and classification processes is artificial, since a classifier often includes the optimization of a feature extraction stage as part of its design.

▶ Classifier Design:-

A classifier is a program that inputs the feature vector and assigns it to one of a set of designated classes or to the “reject” class.

▶ Learning Algorithm:-

A learning algorithm is an algorithm that learns the training data and sets PR from training examples.

▶ Teacher:-

In simple terms a teacher provides learning using training examples termed as supervised learning.

1.1.2 Different Approaches in Pattern Recognition

The different approaches in pattern recognition techniques are discussed below-----

1. Pattern Classification by Decision Functions:-

The main goal of pattern recognition problem is to classify the objects to their respective classes. Decision functions help us in deciding the class to which each object in a system belongs to. Decision functions are two types -

- Linear decision function
- Non-Linear decision function.

Linear decision function

Let $X = (x_1, x_2, \dots, x_n)$ is n number of successive sample feature measurement Where x_n denotes nth feature measurement and X is the pattern vector. Let the values which would be produced by a perfect pattern from the ith class be $w_1^i, w_2^i, \dots, w_n^i$, and let these be the components of the vector W^i . The system will identify the pattern as belonging to the ith class provided below----

$$\Theta. w^i \cdot x > w^j \cdot x \text{ for all } j \neq i$$

Where Θ is a fixed value lying between 0 and 1. The relation between the dot product of two vectors and the cosine of their included angle, we can conclude that the above inequality describe a region of n-dimensional space roughly in the shape of a cone, or more nearly a prism, with vertex at the origin. In this region the standard vector W_i associated with the i th pattern, together with almost all of the vectors which arise from patterns belonging to the i th class are included.

A set of samples with two pattern classes P1 and P2 with a good number of samples in each class we can represent them as given below in $x_1 - x_2$ plane otherwise called as the measurement plane, as shown below. Each sample can be represented as a single point in this plane whose coordinates are given as (x_1, x_2) . The pattern vector of each of the sample is represented in $X_1 - X_2$ plane as $x = (x_1, x_2)'$. If a straight line AB as shown in the figure is drawn, which can be adjusted a little, it is very clear that the two pattern classes can be separated by this straight line.

We represent the line by an equation $d(x) = 0$ with co-efficient as shown below.

$$d(x) = w_1x_1 + w_2x_2 + w_3 = 0$$

We can arrange this in such a way that $d(x) > 0$ for all samples of class 1 and $d(x) < 0$ for all samples of class 2.

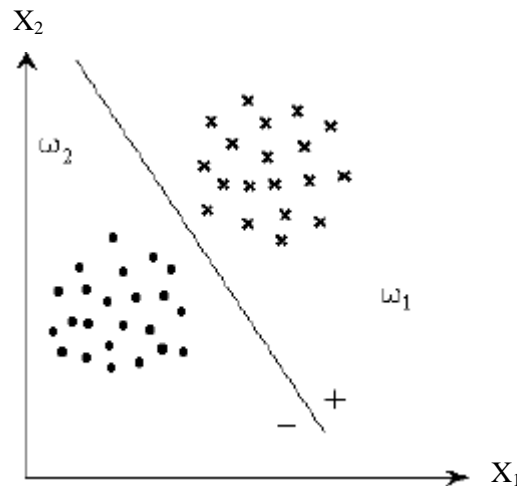


FIG-4

With this line separating class 1 and class 2, any new pattern which is known to be either in class 1 or in class 2 can easily be assigned to either of the two classes by calculating the value of $d(x)$. If the calculated value of $d(x) > 0$ it indicates that the new pattern belongs to class 1 and if $d(x) < 0$ the new pattern can be assumed to belong to class 2.

$$d(x) = w_1x_1 + w_2x_2 + w_3$$

is called a decision function because it helps in deciding the actual class to which each sample belong to. This decision function separates two classes C1 and C2. As the decision function in the above case is a linear function, it is called linear decision function.

Non-Linear decision function

If $d(x) = 1 - x_1^2 - x_2^2 = 0$ then the decision function becomes a non linear function.

$d(x) < 0$ implies the sample belongs to C1

$d(x) > 0$ implies the sample belongs to C2

2. Pattern Classification by Distance Functions:-

In pattern recognition like decision function distance function also takes an important role in pattern classification. Basically in here based on minimal distances of patterns with pattern classes we can classify the pattern class of input patterns. As the classification is based on the minimum distance calculation, this method is called minimum distance classification procedure. The closeness of an incoming pattern to patterns of the possible pattern classes provides a measure in determining the pattern class of the pattern. Let there be m pattern classes in R^n denoted by C1, C2,....., Cm which are represented by the single prototype vectors y_1, y_2, \dots, y_m . The distance between an incoming pattern x and the prototype vectors are-----

$$D_i = \|x - y_i\| = ((x - y_i)^T (x - y_i))^{1/2}, 1 \leq i \leq m$$

The minimum distance classifier will classify x at Cj for which Dj is minimum.

$$D_j = \min \|x - y_i\|, 1 \leq i \leq m$$

Now minimizing D_i^2 by removing xx^T we get,

$$d_i(x) = x^T y_i - 1/2 y_i^T y_i$$

The classifier will classify the sample x as belonging to Ci if $d_i(x) > d_j(x)$

If there are two pattern classes, the decision boundary using minimum distance classification is

$$d_{12}(x) = d_1(x) - d_2(x) = x^T(y_1 - y_2) - 1/2 y_1^T y_1 + 1/2 y_2^T y_2 = 0$$

This describes a hyperplane in normal direction to the vector $y_1 - y_2$. The decision boundary described by the previous equation is a hyperplane which is perpendicular to the vector connecting the two prototypes and bisect.

3. Pattern Classification by Likelihood Functions:-

Let us consider an M -class problem with feature vectors distributed according to $p(x | w_i), i=1,2,\dots,M$. We assume that these likelihood functions are given in a parametric form and that the corresponding parameters form the vectors θ_i which are unknown. To show the dependence on θ_i we write $p(x | w_i, \theta_i)$. Here the goal is to estimate the unknown parameters using a set of known feature vectors in each class. If we further assume that data from one class do not affect the parameter estimation of the others. Let the samples $D = \{x_1, x_2, \dots, x_n\}$ drawn from probability density function (pdf) $p(x; \theta)$ where The aim is to estimate the value of the parameter vector θ based on the training samples D . We assume that the training samples are occurrences of independent and identically distributed random variables distributed according to the density $p(x | \theta)$. The maximum likelihood estimate or the maximum likelihood estimate θ' maximizes the probability of the training data with respect to θ . Due to the independent and identically distributed assumption, the probability of D is-----

$$p(D | \theta) = p(x_1, x_2, \dots, x_n | \theta) = \prod_{i=1}^n p(x_i | \theta)$$

This is a function of θ and it is also known as the likelihood function of θ with respect to D .

4. Trainable Pattern Classifier- The Deterministic Approach:-

Trainable classifiers use a deterministic approach to define their decision functions. As with other classifiers a formula exists to define the form of decision boundary. But with trainable classifiers they learn the value of their coefficients from the respective samples of their classes. An assumption is made that the features selected are sufficient to determine an unique boundary of n space that would define all the classes. The deterministic approach relies only on individual samples. Here the statistical nature of samples is not used. The coefficients for the assumed function are calculated by techniques such as iteratively trying to minimize all sample to boundary line distances. A common trainable classifier is the Perceptron algorithm.

5. Trainable Pattern Classifier- The Statistical Approach:-

The statistical approach uses Bayesian decision functions in its classification. This Bayesian decision function reduces probability of an error. The threshold boundary is defined here as:-

$$P(x | A) P(A) = P(x | B) P(B)$$

According to Bayes theorem:-

$$P(x | A) = P(A | x) P(x) / P(A)$$

So we can redefine the first equation as:-

$$P(A | x) = P(B | x)$$

Bayes decisions functions

$$d_i(x) = p(w_i/x) \quad i=1, \dots, M$$

minimizes the average loss of misclassification and yields the lowest error probability. For two class case

$$d(x) = d_1(x) - d_2(x) = 2 p(w_1/x) - 1$$

$$x \rightarrow w_1 \quad \text{if } p(w_1/x) > 0.5$$

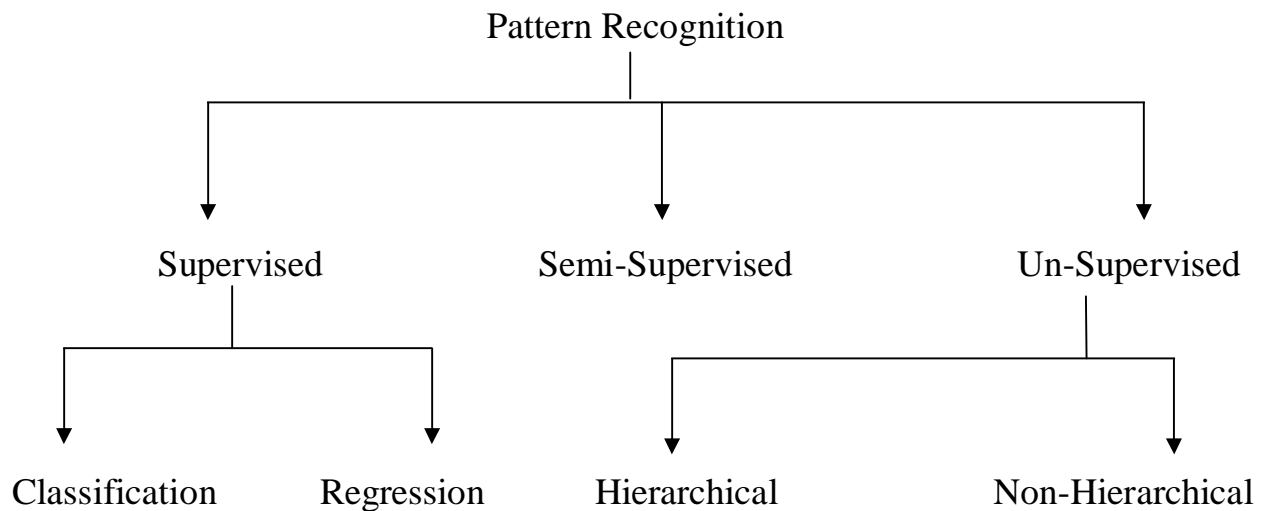
$$x \rightarrow w_2 \quad \text{if } p(w_2/x) < 0.5$$

Robbin-Maros algorithm [96] is an example of Statistical Approach.

6. Syntactic Pattern Recognition:-

Syntactic pattern recognition not only represent the pattern but also its structure within which pattern occurs. These use features to build pattern primitives. The primitives combine structurally to create a pattern. This structure is explicitly defined. These are only applicable to pattern recognition problems. In some classification problems no relationship exists between features. Example is speech recognition. Another example of domain where syntactic PR is used is scene analysis.

1.1.3 Pattern Recognition Methods



1.1.3.1 Supervised Method

- ▶ Supervised learning is the machine learning task of inferring a function from supervised training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which is called a classifier (if the output is discrete, see classification) or a regression function (if the output is continuous, see regression).
- ▶ In **supervised learning** one is furnished with input (x_1, x_2, \dots) and output (y_1, y_2, \dots) and are challenged with finding a function that approximates this behavior in a generalize fashion. The output could be a class label (in classification) or a real number (in regression) -- these are the "supervision" in supervised learning.
- ▶ The goal is to learn a mapping from x to y , given a training set made of pairs (x_i, y_i) . Here $y_i \in y$ is called the labels or targets of the example x_i . If the labels are numbers $y = (y_i)_{i \in n}^T$ denotes the column vector of labels. Again a standard requirement is that the pairs (x_i, y_i) is sampled from some distribution which here ranges over $x \times y$.
- ▶ Supervision: The training data (observations, measurements, etc.) are accompanied by labels indicating the class of the observations. New data is classified based on the training set.

Steps in Supervised Learning

1. Determine the type of training examples. Before doing anything else, the user should decide what kind of data is to be used as a training set. In the case of handwriting analysis, for example, this might be a single handwritten character, an entire handwritten word, or an entire line of handwriting.
2. Gather a training set. The training set needs to be representative of the real-world use of the function. Thus, a set of input objects is gathered and corresponding outputs are also gathered, either from human experts or from measurements.
3. Determine the input feature representation of the learned function. The accuracy of the learned function depends strongly on how the input object is represented. Typically, the input object is transformed into a feature vector, which contains a number of features that are descriptive of the object. The number of features should not be too large, because of the curse of dimensionality; but should contain enough information to accurately predict the output.
4. Determine the structure of the learned function and corresponding learning algorithm. For example, the engineer may choose to use support vector machines or decision trees.
5. Complete the design. Run the learning algorithm on the gathered training set. Some supervised learning algorithms require the user to determine certain control parameters.

These parameters may be adjusted by optimizing performance on a subset (called a *validation* set) of the training set, or via cross-validation.

6. Evaluate the accuracy of the learned function. After parameter adjustment and learning, the performance of the resulting function should be measured on a test set that is separate from the training set.

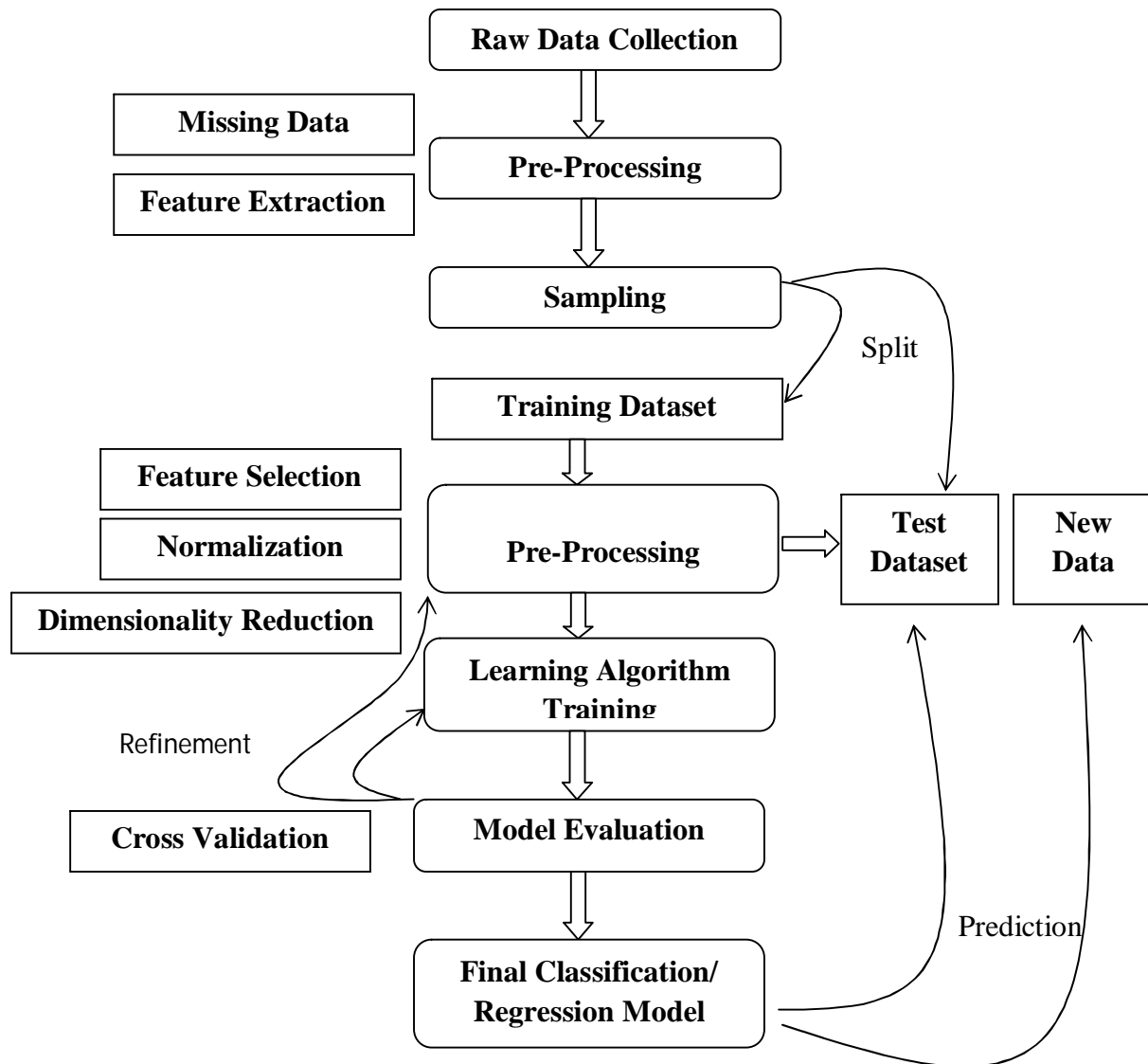


FIG-5: Flow Diagram of the Supervised Learning

Example of Supervised Learning Algorithm

- Decision Tree Learning
- Bayesian Learning
- Learning using Artificial Neural Network
- Back propagation
- Support Vector Machine
- Nearest Neighbor Learning
- Probably approximately correct learning (PAC) learning
- Random Forest

1.1.3.2 Semi-supervised Method

Semi-supervised learning is actually a supervised method that avoids labeling a large number of instances. This is done by using some of the labeled data to help the classifier labeling the unlabeled data. Then, this automatic labeled data is also used by the training process. Another supervised method that helps mining labeled data is called active learning. Basically, it decides which data should be labeled to improve the classifier performance with less data. These two options are really interesting, as they have the benefits of both supervised and unsupervised learning: interactivity and taking advantage of unlabeled data. With a few labeled instances and the great amount of unlabeled images at our disposal this system could perform well for shot type detection.

Semi-supervised learning is a halfway between supervised and unsupervised learning. In addition to unlabeled data, the algorithm is provided with some supervision information but not necessary for all examples. Often this information will be the targets associated with some of the examples. In this case the data set $X = (x_i)_{i \in n}$ can be divided into two parts $X_l := (x_1, x_2, \dots, x_i)$ for which labels $Y_l := (y_1, y_2, \dots, y_i)$ are provided, and the points $X_u := (x_{i+1}, \dots, x_{i+u})$, the labels of which are not known.

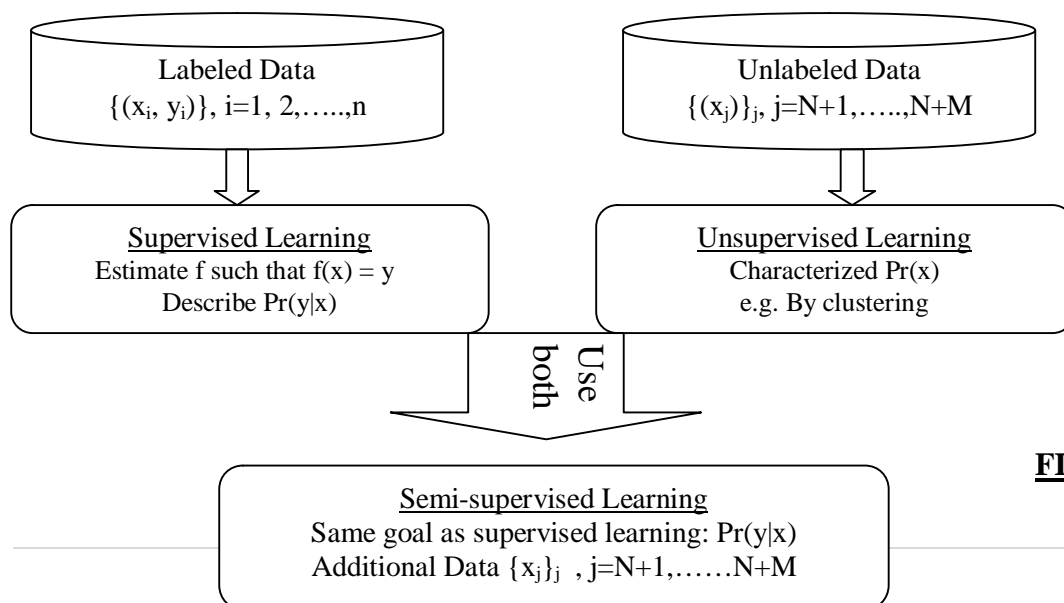


FIG-6

1.1.3.3 Unsupervised Method

Always in pattern recognition tasks the training data of known class labels are not available like supervised learning method. In this type of problem, we are given a set of feature vectors \mathbf{x} and the goal is to unravel the underlying similarities and cluster (group) “similar” vectors together. This is known as unsupervised pattern recognition or unsupervised learning or clustering.

Unsupervised learning is the machine learning task of inferring a function to describe hidden structure from unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning.

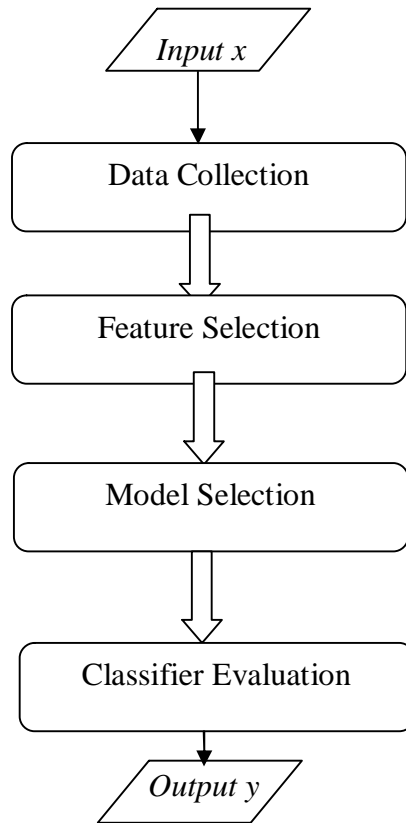
Unsupervised learning shows how systems can learn to represent particular input patterns in a way that reflects the statistical structure of the overall collection of input patterns. It is distinguished from supervised learning (and reinforcement learning) in that the learner is given only unlabeled examples.

So far, in all the learning techniques we considered, a training example consisted of a set of attributes (or features) and either a class (in the case of classification) or a real number (in the case of regression) attached to it. Unsupervised Learning takes as training examples the set of attributes/features alone. The main purpose of unsupervised learning is to attempt to find natural partitions in the training set.

Let $X = (x_1, x_2, \dots, x_n)$ be a set of n examples where $x_i \in X$. Typically it is assumed that the points are drawn from a common distribution on X . The goal of the unsupervised learning is to find interesting structure in the data X .

Why Unsupervised Learning is Useful?

- ▶ Collecting and labeling a large set of sample patterns can be very expensive. By designing a basic classifier with a small set of labeled samples, and then tuning the classifier up by allowing it to run without supervision on a large, unlabeled set, much time and trouble can be saved.
- ▶ Training with large amounts of often less expensive, unlabeled data, and then using supervision to label the groupings found. This may be used for large "data mining" applications where the contents of a large database are not known beforehand.
- ▶ Unsupervised methods can also be used to find features which can be useful for categorization. There are unsupervised methods that represent a form of data-dependent "smart pre-processing" or "smart feature extraction."
- ▶ Lastly, it can be of interest to gain insight into the nature or structure of the data. The discovery of similarities among patterns or of major departures from expected characteristics may suggest a significantly different approach to designing the classifier.



Flow Diagram of Unsupervised Method

Unsupervised Techniques can be divided into 2 types of techniques:-

1. Hierarchical – Here the number of clusters are fixed [97]. This fixed number is not known.

Hierarchical clustering follows one of two approaches:-

- I. Agglomerative methods start with each observation as a cluster and with each step combine observations to form clusters until there is only one large cluster.
- II. Divisive methods begin with one large cluster and proceed to split into smaller clusters items that are most dissimilar.

There are five ways of defining inter-cluster distance:

- a) single linkage (based on the shortest distance between objects);
- b) complete linkage (based on the largest distance between objects);
- c) average linkage (based on the average distance between objects);
- d) Ward's method (based on the sum of squares between the two clusters, summed over all variables)

- e) Centroid method (based on the distance between clusters Centroid). Here the distance between clusters is that between their Centroid (mean vectors).

2. Non-Hierarchical:- Here the no of clusters are fixed but that value may be known or not known. The Nonhierarchical clustering is the partitioning of the sample. The mechanisms used here are:-

Each cluster has a seed point and all objects within a prescribed distance are included in that cluster. Another way of nonhierarchical clustering is to loop through the sample, assigning each case to the seed point to which it is closest.

Nonhierarchical clustering has three approaches:

1) Sequential threshold :-This is based on one cluster seed at a time and membership in that cluster fulfilled before another seed is selected, i.e., looping through all n points before updating the seeds, as in the K-MEANS procedure

2) Parallel threshold :-This is based on simultaneous cluster seed selection and membership threshold distance adjusted to include more or fewer objects in the clusters, i.e., updating the seeds , as in the ISODATA procedure)

3) Optimizing: - This is same as the others except it allows for reassignment of objects to another cluster based on some optimizing criterion).

Some Important Application Area of Unsupervised Learning:-

- Bio-Medical Image Processing
- Image Segmentation
- Computational Biology and Bio-Informatics
- QSAR and molecular modeling in Chemo informatics
- Market Research
- Customer Segmentation
- Social Network Analysis
- Sippy map optimization
- Crime Analysis

1.2 Introduction to Soft Computing

Soft computing is a consortium of methodologies that works synergistically and provides in one form or another flexible information processing capability for handling real life ambiguous situations. Soft Computing is a term applied to a field within computer science which is characterized by the use of inexact solutions to computationally-hard tasks such as the solution of NP-complete problems, for which an exact solution cannot be derived in polynomial time. Every computing process that purposely includes imprecision into the calculation on one or more

levels and allows this imprecision either to change the granularity of the problem, or to “soften” the goal of optimization at some stage, is defined as to belonging to the field of soft computing. Soft Computing is a term used in computer science to refer to problems in computer science whose solutions are unpredictable, uncertain and between 0 and 1. Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation.

In effect, the role model for soft computing is the human mind. In 1994 Zadeh [98] first defined “Soft Computing”, the currently handled concepts used to be referred to in an isolated way, whereby each was spoken of individually with an indication of the use of fuzzy methodologies. Zadeh defined the Soft Computing in the following way:-

“Basically Soft Computing is not a homogeneous body of concepts and techniques. Rather it is a partnership of distinct methods that in one way or another confirm to its guiding principle. At this juncture, the dominant aim of the soft computing is to exploit the tolerance for imprecision and uncertainty to achieve tractability, robustness and low solutions cost. The principle constituents of soft computing are fuzzy logic, Neurocomputing, and probabilistic reasoning, with the latter subsuming genetic algorithms, belief networks, chaotic system, and parts of learning theory. In the partnership of fuzzy logic, Neurocomputing, and probabilistic reasoning, fuzzy logic is mainly concerned with imprecision and approximate reasoning; Neurocomputing with learning and curve fitting; and probabilistic reasoning with uncertainty and belief propagation.” Components of Soft Computing includes-

1. Artificial Neural Network
2. Fuzzy Logic
3. Genetic Algorithm

Some Important Application Area of Soft Computing

- Handwritten Character Recognition
- Image Processing
- Data Compression
- Architecture
- Decision Support System
- Power System

1.2.1 Artificial Neural Network (ANN)

A neural network is an interconnected assembly of simple processing elements, units or nodes, operating in parallel which can acquire, store, and utilize experiential knowledge. The processing ability of the network is stored in the inter unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

A neuron (or nerve cell) is a special biological cell that processes information (see **FIG-7**). It is composed of a cell body, or soma, and two types of out-reaching tree-like branches: the axon and the dendrites. The cell body has a nucleus that contains information about hereditary traits and plasma that holds the molecular equipment for producing material needed by the neuron. A neuron receives signals (impulses) from other neurons through its dendrites (receivers) and transmits signals generated by its cell body along the axon (transmitter), which eventually branches into strands and sub strands. At the terminals of these strands are the synapses. A synapse is an elementary structure and functional unit between two neurons (an axon strand of one neuron and a dendrite of another). When the impulse reaches the synapse's terminal, certain chemicals called neurotransmitters are released. The neurotransmitters diffuse across the synaptic gap, to enhance or inhibit, depending on the type of the synapse, the receptor neuron's own tendency to emit electrical impulses. The synapse's effectiveness can be adjusted by the signals passing through it so that the synapses can learn from the activities in which they participate. This dependence on history acts as a memory, which is possibly responsible for human memory.

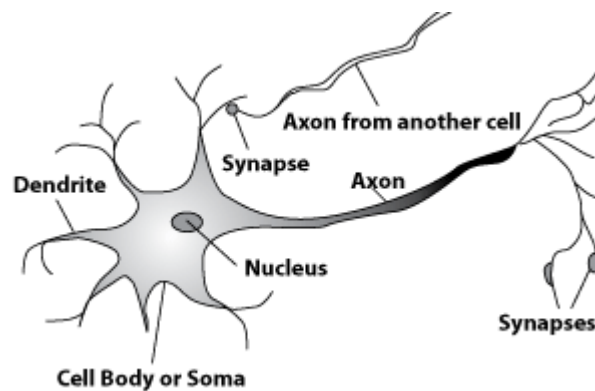


FIG-7: SCHEMATIC DIAGRAM OF A BIOLOGICAL NEURON

Motivated by architecture and functionalities of biological neuron researchers have modeled the Artificial Neural Network (ANN). We may identify the basic elements of the neuron model. These are stated below.

1. A set of synapses, each of which is characterized by a weight or strength of its own. Specifically, a signal x_j at the input of synapse j connected to neuron k is multiplied by the synaptic weight w_{kj} . It is important to make a note of the manner in which the subscripts of the synaptic weight w_{kj} are written. The first subscript refers to the neuron in question and the second subscript refers to the input end of the synapse to which the weight refers. The weight w_{kj} is positive if the associated synapse is excitatory; it is negative if the synapse is inhibitory.

2. An adder for summing the input signals, weighted by the respective synapses of the neuron.
3. An activation function for limiting the amplitude of the output of a neuron. The activation function is also referred to in the literature as a squashing function in that it squashes (limits) the permissible amplitude range of the output signal to some finite value. Typically, the normalized amplitude range of the output of a neuron is written as the closed unit interval $[0, 1]$ or alternatively $[-1, 1]$.

In mathematical terms, we may describe a neuron k by writing the following pair of equations:

$$net_j = \sum w_{ij}x_j ,$$

$$O_j = \phi (net_j) \quad \begin{matrix} 1, & \text{if } net_j > \theta_j \\ 0, & \text{elsewhere} \end{matrix}$$

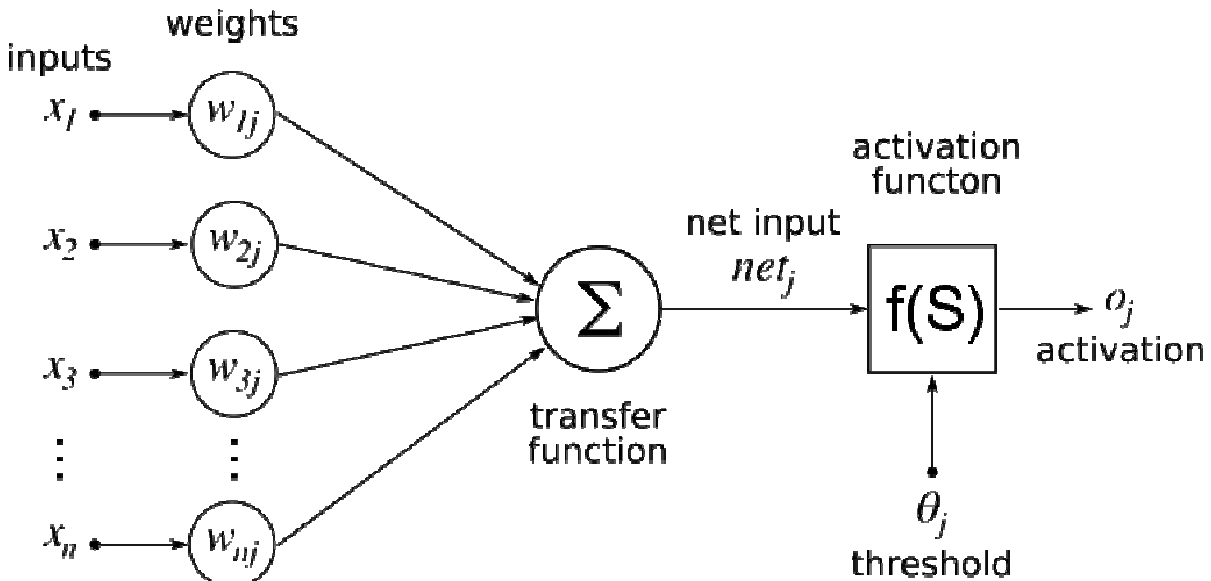


FIG-8

Standard structure of an artificial neural network

Artificial neural network consists of three groups, or layers, of units: a layer of "**input**" units is connected to a layer of "**hidden**" units, which is connected to a layer of "**output**" units.

- Input units
 - represents the input as a fixed-length vector of numbers (user defined)
- Hidden units
 - calculate thresholded weighted sums of the inputs

- represent intermediate calculations that the network learns
- Output units
 - represent the output as a fixed length vector of numbers

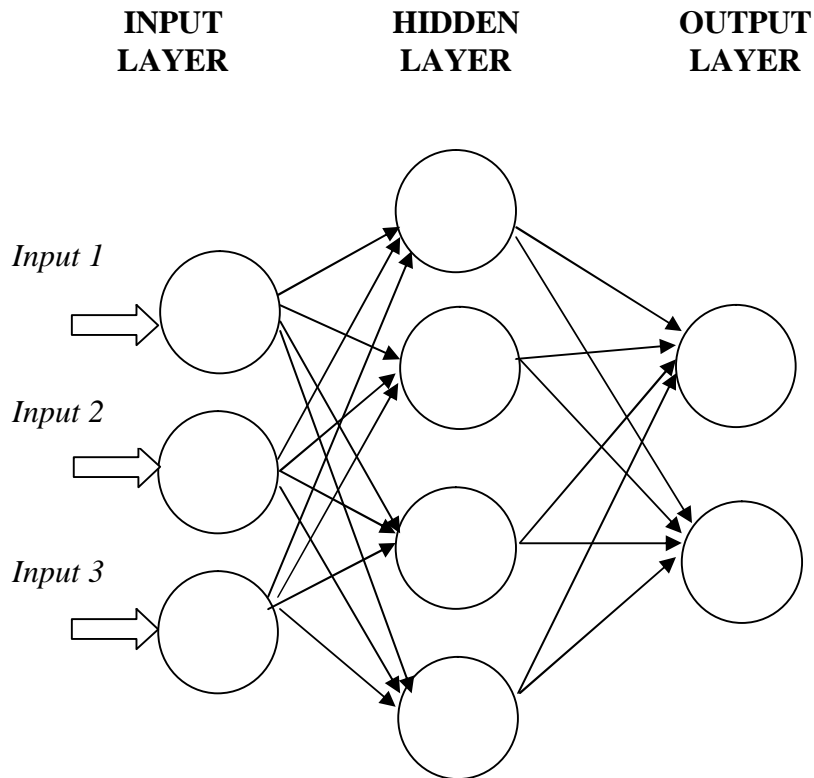


FIG-9: SCHEMATIC STRUCTURE OF AN ANN

A geometric interpretation (adopted and modified from Lippmann [99]) shown in **FIG-10** can help explicate the role of hidden units (with the threshold activation function). Each unit in the first hidden layer forms a hyperplane in the pattern space; boundaries between pattern classes can be approximated by hyperplane. A unit in the second hidden layer forms a hyper region from the outputs of the first-layer units; a decision region is obtained by performing an AND operation on the hyperplane. The output-layer units combine the decision regions made by the units in the second hidden layer by performing logical OR operations.


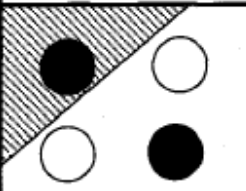

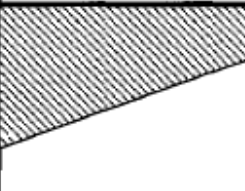
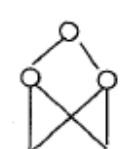
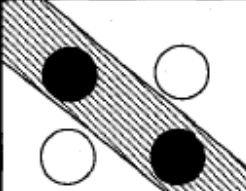

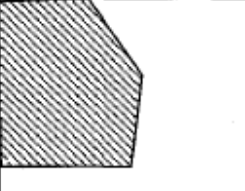
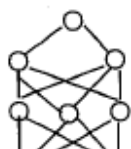
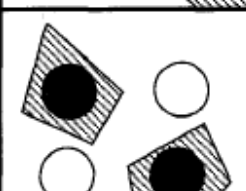


Structure	Description of decision regions	Exclusive-OR problem	Classes with meshed regions	General region shapes
 Single layer	Half plane bounded by hyperplane			
 Two layer	Arbitrary (complexity limited by number of hidden units)			
 Three layer	Arbitrary (complexity limited by number of hidden units)			

FIG-10: A GEOMETRIC INTERPRETATION OF THE ROLE OF HIDDEN LAYER

ANN can be viewed as weighted directed graphs in which artificial neurons are nodes and directed edges (with weights) are connections between neuron outputs and neuron inputs. Based on the connection pattern (architecture), ANN can be grouped into two categories (FIG-11):--

- Feed-forward networks, in which graphs have no loops,
- Recurrent (or *feedback*) networks, in which loops occur because of feedback connections.

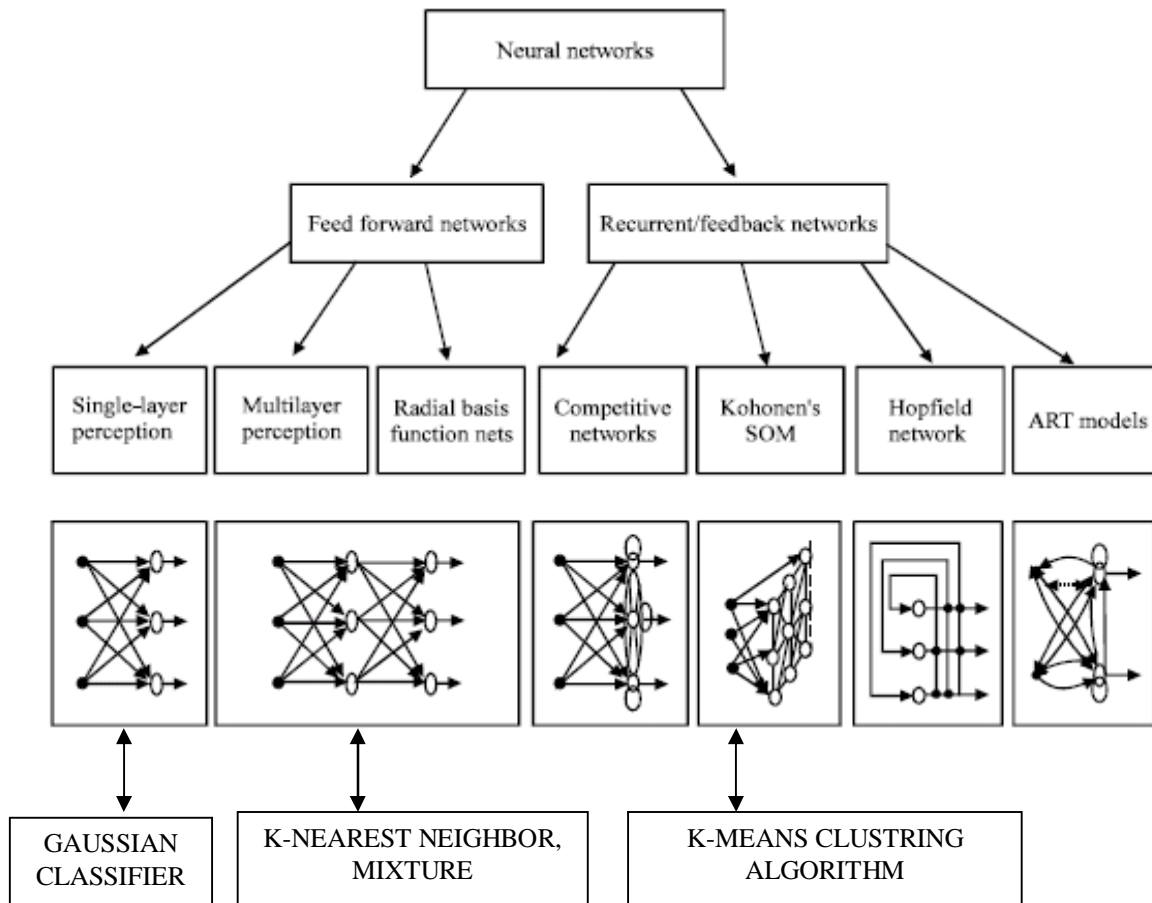


FIG-11: TAXONOMY OF NEURAL NETS

Applications off ANN

► **Classification**

- In marketing:- consumer spending pattern classification
- In defence:- radar and sonar image classification
- In agriculture & fishing:- fruit and catch grading
- In medicine:- ultrasound and electrocardiogram image classification, EEGs, medical diagnosis

► **Recognition and identification**

- In general computing and telecommunications:- speech, vision and handwriting recognition
- In finance:- signature verification and bank note verification

- ▶ **Assessment**
 - In engineering:- product inspection monitoring and control
 - In defence:- target tracking
 - In security:- motion detection, surveillance image analysis and fingerprint matching
- ▶ **Forecasting and prediction**
 - In finance:- foreign exchange rate and stock market forecasting
 - In agriculture:- crop yield forecasting
 - In marketing:- sales forecasting
 - In meteorology:- weather prediction

1.2.2 Fuzzy Logic

Fuzzy logic can be defined as a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth - truth values between “completely true” and “completely false”. The notion of fuzzy systems is that truth values (in fuzzy logic) or membership values (in fuzzy sets) are indicated by a value on the range [0.0, 1.0], with 0.0 representing absolute Falseness and 1.0 representing absolute Truth. Fuzzy logic is an extension of Boolean logic by Lot_ Zadeh in 1965 based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory. By introducing the notion of degree in the verification of a condition, thus enabling a condition to be in a state other than true or false, fuzzy logic provides a very valuable flexibility for reasoning, which makes it possible to take into account inaccuracies and uncertainties. A fuzzy logic system (FLS) can be defined as the nonlinear mapping of an input data set to a scalar output data [100]. A FLS consists of four main parts: fuzzifier, rules, inference engine, and defuzzifier.

Crisp Vs. Fuzzy Set

The invention of the fuzzy set was motivated by the need to capture and represent the real world with its fuzzy data due to its uncertainty. Uncertainty can be caused by vagueness in the language objects and situations or imprecision in the measurement. Crisp set theory is not capable of representing those descriptions and classifications in many cases. Instead of avoiding uncertainty a set theory was developed by Lofti Zadeh that captures this uncertainty. Unlike crisp set here membership function denotes, basically the degree of membership denotes the probability of presence of elements within or without the set. In the fuzzy theory, fuzzy set A of universe of discourse X is defined by function $\mu_A(x)$ called the membership function of set A .

$$\begin{aligned} \mu_A(x) : X \rightarrow [0,1], \text{ where } \mu_A(x) &= 1 \text{ if } x \text{ is totally in } A; \\ \mu_A(x) &= 0 \text{ if } x \text{ is not in } A; \\ 0 < \mu_A(x) < 1 & \text{ if } x \text{ is partly in } A \end{aligned}$$

Let X be the universe of discourse and its elements be denoted as x . In the classical set theory, crisp set A of X is defined as function $f_A(x)$ called the characteristic function of A .

$$f_A(x): X \rightarrow \{0,1\}, \text{ where } f_A(x) = \begin{cases} 1, & \text{if } x \in A \\ 0, & \text{if } x \notin A \end{cases}$$

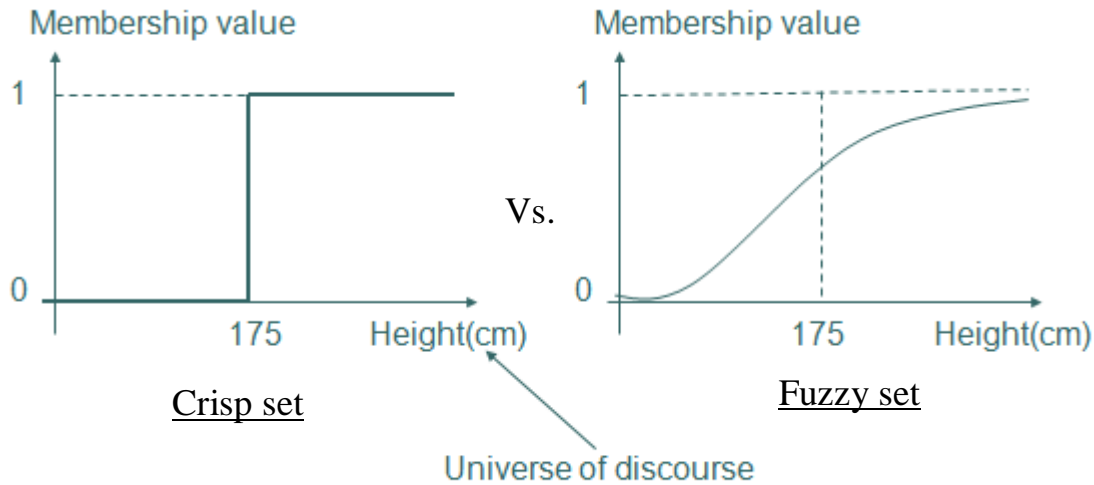


FIG-12: A Comparison of Membership Function Crisp vs. Fuzzy Set

How Fuzzy Logic Works?

► **Step 1:-**

- Determine input and output relationships.
- Determine the least number of variables for inputs to the fuzzy logic system.

► **Step 2:-**

- Break down the control problem into a series of IF X AND Y, THEN Z rules based on the fuzzy logic rules.
- These IF X AND Y, THEN Z rules should define the desired system output response for the given systems input conditions.

► **Step 3:-**

- Create a fuzzy logic membership function that defines the meaning or values of the input and output terms used in the rules.

► **Step 4 :-**

- After the membership functions are created, program everything then into the fuzzy logic system.

► **Step 5 :-**

- Finally, test the system, evaluate results and make the necessary adjustments until a desired result is obtain.

The above steps are summarized into below Algorithm that has mainly three step process.

I. Fuzzification

Firstly, a crisp set of input data are gathered and converted to a fuzzy set using fuzzy linguistic variables, fuzzy linguistic terms and membership functions.

II. Evaluation of Rules

Afterwards, an inference is made based on a set of rule. This is the application of the fuzzy logic rules.

III. Diffuzzification

Lastly, the resulting fuzzy output is mapped to a crisp output using the membership functions, in the diffuzzification step.

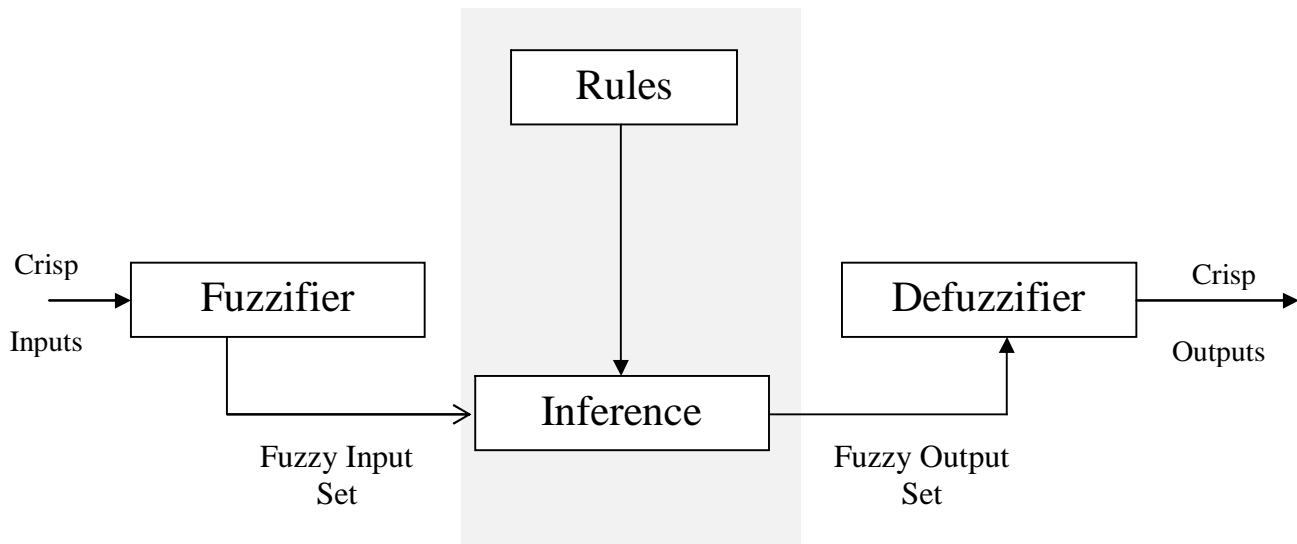


FIG-13: A Fuzzy Logic System

.....

Algorithm: - Fuzzy logic algorithm

.....

1. Define the linguistic variables and terms (initialization).
2. Construct the membership functions (initialization).
3. Construct the rule base (initialization).
4. Convert crisp input data to fuzzy values using the membership functions (fuzzification).
5. Evaluate the rules in the rule base (inference).
6. Combine the results of each rule (inference).
7. Convert the output data to non-fuzzy values (defuzzification).

1.2.3 Genetic Algorithm

A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems. Genetic algorithm (GA) is a search and optimization method which works by mimicking the evolutionary principles and chromosomal processing in natural genetics.

A GA begins its search with a random set of solutions usually coded in binary strings. Every solution is assigned a fitness which is directly related to the objective function of the search and optimization problem. Thereafter, the population of solutions is modified to a new population by applying three operators similar to natural genetic operators-reproduction, crossover, and mutation. It works iteratively by successively applying these three operators in each generation till a termination criterion is satisfied. Over the past decade and more, GA has been successfully applied to a wide variety of problems, because of their simplicity, global perspective, and inherent parallel processing. GAs implemented as computer simulation in which a population of abstract representations (called chromosomes or the genotype of the genome) of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem evolves towards better solutions. Traditionally, solutions are represented in binary's string of 0s and 1s but other encoding is also possible.

Why Genetic Algorithms (GAs)?

- 1) Evolution produced good individuals might work for solving complex problems.
- 2) Most real life search and optimization problems cannot be solved in polynomial amount of time using any deterministic algorithm.
- 3) Near optimal solutions requiring less time more desirable than optimal solutions with huge amount of time.

Various Steps involved in GA Procedure

1. *Initialization:* - The initial population of candidate solutions is usually generated randomly across the search space. However, domain-specific knowledge or other information can be easily incorporated.
2. *Evaluation:-* Once the population is initialized or an offspring population is created, the fitness values of the candidate solutions are evaluated.
3. *Selection:-* Selection allocates more copies of those solutions with higher fitness values and thus imposes the survival-of-the-fittest mechanism on the candidate solutions. The main idea of selection is to prefer better solutions to worse ones, and many selection procedures have been proposed to accomplish this idea, including roulette-wheel selection, stochastic universal selection, ranking selection and tournament selection, some of which are described in the next section.
4. *Recombination:* - Recombination combines parts of two or more parental solutions to create new, possibly better solutions (i.e. offspring). There are many ways of accomplishing this (some of which are discussed in the next section), and competent performance depends on a properly designed recombination mechanism. The offspring under recombination will not be identical to any particular parent and will instead combine parental traits in a novel manner (Goldberg, 2002).
5. *Mutation:-* While recombination operates on two or more parental chromosomes, mutation locally but randomly modifies a solution. Again, there are many variations of mutation, but it usually involves one or more changes being made to an individual's trait or traits. In other words, mutation performs a random walk in the vicinity of a candidate solution.
6. *Replacement:* - The offspring population created by selection, recombination, and mutation replaces the original parental population. Many replacement techniques such as elitist replacement, generation-wise replacement and steady-state replacement methods are used in GAs.
7. Repeat steps 2–6 until a terminating condition is met.

Applications off Genetic Algorithm

- Dynamic process control
- Induction of rule optimization
- Discovering new connectivity topologies
- Simulating biological models of behavior and evolution
- Complex design of engineering structures
- Pattern recognition

- Scheduling
- Transportation
- Layout and circuit design
- Telecommunication
- Graph-based problems

1.3 Application Area of Pattern Recognition Technique

I. Man-machine communication:

- a. Automatic speech recognition.
- b. Speaker identification
- c. OCR systems.
- d. Cur sine script recognition
- e. Image understanding .

II. Biomedical application:

- a. ECG, EEG, EMG analysis.
- b. Cytological, histological & other stereological applications.
- c. X-ray analysis.
- d. Medical diagnostics.

III. Application in physics:

- a. High energy physics.
- b. Bubble chamber & other forms of track analysis.

IV. Crime & criminal detection:

- a. Fingerprint.
- b. Hand writing.
- c. Speech & sound.
- d. Photographs.

V. Natural resources study & estimation:

- a. Agriculture.
- b. Hydrology.
- c. Forestry.
- d. Geology.
- e. Environment.
- f. Cloud pattern.
- g. Urban quality.

VI. Stereological applications:

- a. Metal detection.

- b. Mineral processing.
- c. Biology.

VII. Military applications:

- a. Detection of nuclear explosions.
- b. Missile guidance & detection.
- c. Radar & sonar signal detection.
- d. Target identification.
- e. Naval submarine detection, etc.

Chapter 2:

CLUSTERING TECHNIQUES

2.1 Introduction to Clustering

Clustering is the process of extracting the “natural groups” present in a given data set and each such group is termed as a cluster. The objects belonging to one cluster are more similar and objects belonging to different clusters are dissimilar. Clustering is an unsupervised method for grouping similar type objects from heterogeneous collection of data points and discover the inherent structure present in the data set that plays an important role in data mining [1]. For a given data set $S = \{X_1, X_2, \dots, X_n\} \in \mathcal{R}^m$ what one perceives to be the groups present in S by viewing the scatter diagram of S , is termed as natural groups of S . A. Ben-Hur [2] proposed a method to detect the clusters present in the data set based on “natural grouping”.

Let the set of n patterns $S = \{x_1, x_2, \dots, x_n\}$ and K clusters are represented by C_1, C_2, \dots, C_K

1. $C_i \neq \phi$, for $i = 1, 2, \dots, K$
2. $C_i \cap C_j = \phi$ for $i \neq j$ and
3. $\bigcup_{i=1}^K C_i = S$ where ϕ represents null set.

A clustering algorithm attempts to find natural groups of components (or data) based on some similarity. Also, the clustering algorithm finds the centroid of a group of data sets. To determine cluster membership, most algorithms evaluate the distance between a point and the cluster centroids. The output from a clustering algorithm is basically a statistical description of the cluster centroids with the number of components in each cluster.

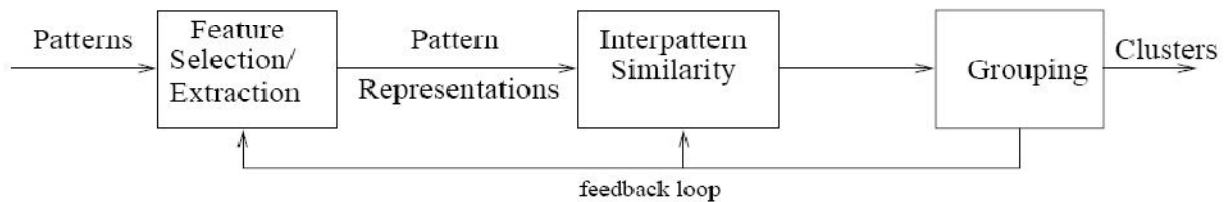
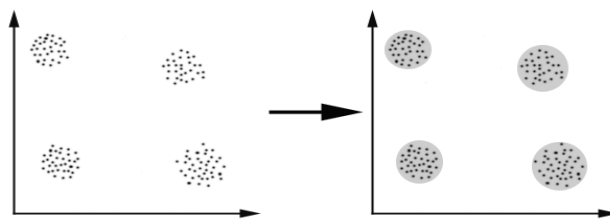


FIG-14: Different stages of Clustering Technique [101]

Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their *similarity*. Clustering can also be used for outlier detection, where outliers (values that are “far away” from any cluster) may be more interesting than common cases.



The clustering algorithm can be categorized as below **FIG-15**.

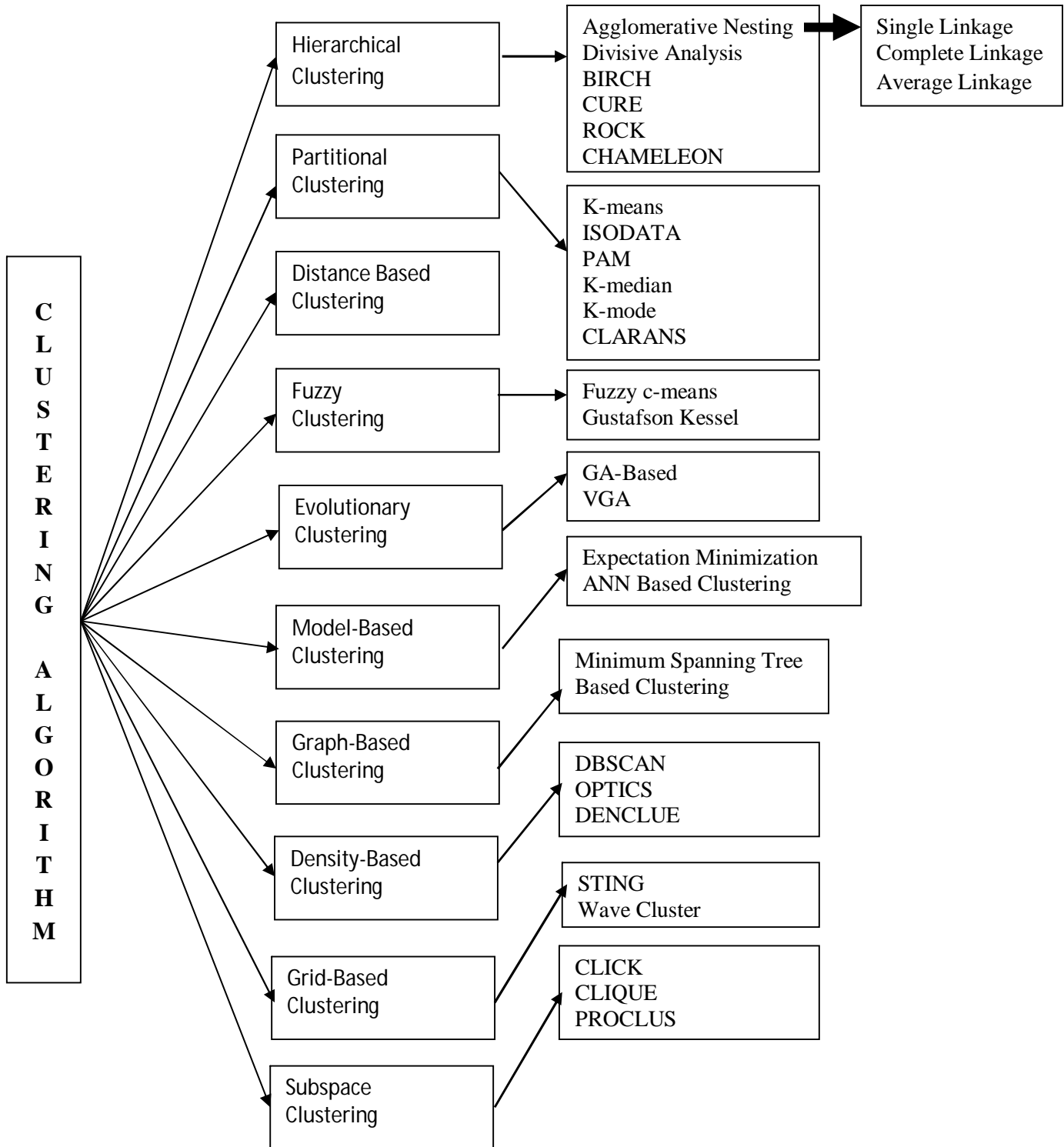


FIG-15: Different type of Clustering Technique

2.1 Clustering Techniques

2.2.1 Hierarchical Clustering

A hierarchical clustering method works by grouping data objects into a tree of clusters. Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion [102]. Here Clusters are created in levels actually creating sets of clusters at each level. Given the input set S , the goal is to produce a hierarchy (dendrogram) in which nodes represent subsets of S . Each level of the tree represents a partition of the input data into several (nested) clusters or groups. The root is the whole input set S . The leaves are the individual elements of S . The internal nodes are defined as the union of their children.

2.2.1.1 Agglomerative Nesting

This is a bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied. Most hierarchical clustering methods belong to this category. The step of following clustering defined by S.C. Johnson in 1967 has been discussed below. If there is given a set of N items to be clustered, and an $N \times N$ distance (or similarity) matrix, the basic process of Agglomerative Nesting is below-----

- 1) Start by assigning each item to a cluster, so that if you have N items, you now have N clusters, each containing just one item. Let the distances (similarities) between the clusters the same as the distances (similarities) between the items they contain.
- 2) Find the closest (most similar) pair of clusters and merge them into a single cluster, so that now you have one cluster less.
- 3) Compute distances (similarities) between the new cluster and each of the old clusters.
- 4) Repeat steps 2 and 3 until all items are clustered into a single cluster of size N .

2.2.1.1.1 Single Linkage Method

The dissimilarity between two clusters is the minimum dissimilarity between members of the two clusters. This method produces long chains which form loose, straggly clusters. Step by step process is discussed below.

(1) By placing each pattern in its own cluster a list of inter pattern distances for all distinct unordered pairs of patterns is constructed, and sort this list in ascending order.

(2) Step through the sorted list of distances, forming for each distinct similarity value d_k a graph on the patterns where pairs of patterns closer than d_k are connected by a graph edge. If all the patterns are members of a connected graph, stop. Otherwise, repeat this step.

$$d_k = \text{sim}(c_i, c_j) = d_{\min}(c_i, c_j) = \min_{x \in c_i, y \in c_j} \|x - y\|$$

(3) The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a partition (clustering) identified by simply connected components in the corresponding graph.

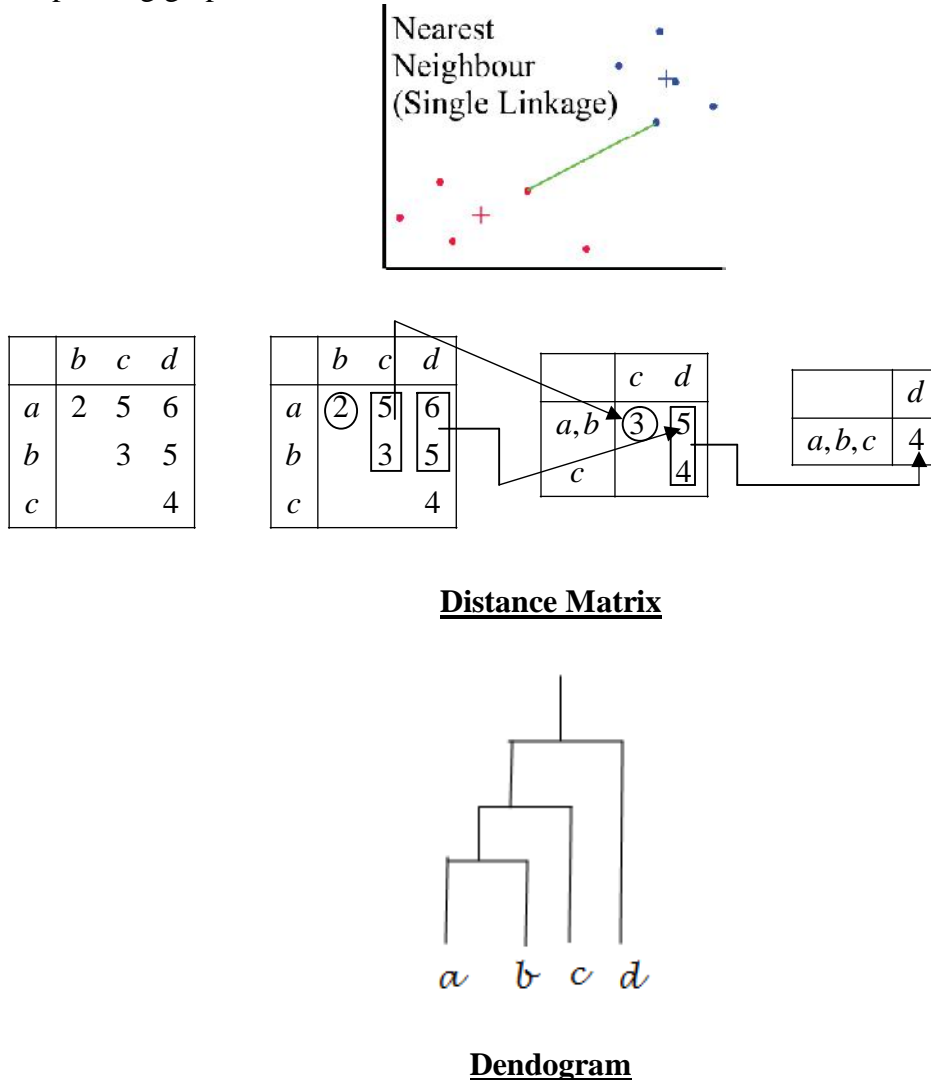


FIG-16: Graphical Example of Single Linkage Method

2.2.1.1.2 Complete Linkage Method

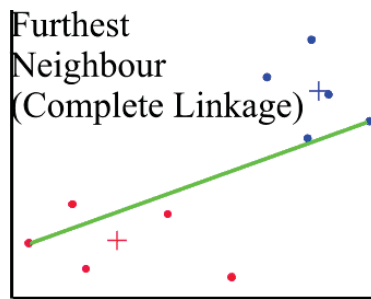
Here the distance between one cluster and another cluster to be equal to the longest distance from any member of one cluster to any member of the other cluster. The dissimilarity between two groups is equal to the greatest dissimilarity between a member of cluster i and a member of cluster j . This method tends to produce very tight clusters of similar cases. Step by step process has been illustrated below.

(1) By placing each pattern in its own cluster a list of inter pattern distances for all distinct unordered pairs of patterns is constructed, and sort this list in ascending order.

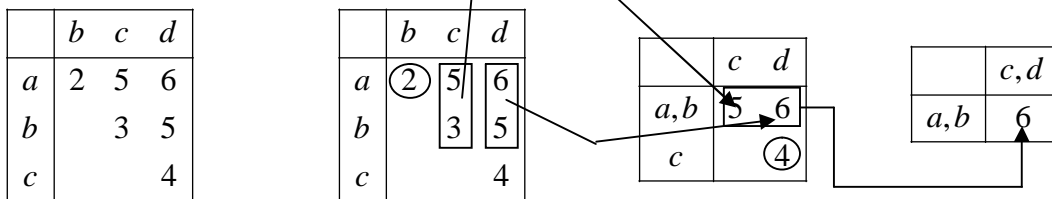
(2) Step through the sorted list of distances, forming for each distinct similarity value d_k a graph on the patterns where pairs of patterns closer than d_k are connected by a graph edge. If all the patterns are members of a completely connected graph, stop.

$$d_k = \text{sim}(c_i, c_j) = d_{\max}(c_i, c_j) = \max_{x \in c_i, y \in c_j} \|x - y\|$$

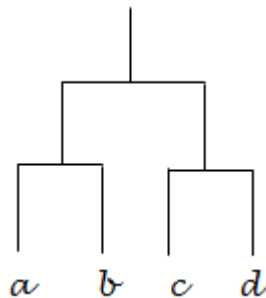
(3) The output of the algorithm is a nested hierarchy of graphs which can be cut at a desired dissimilarity level forming a partition (clustering) identified by completely connected components in the corresponding graph.



If we take the previous example to illustrate the complete linkage method then dendrogram is in below figure-----



Distance Matrix



Dendrogram

FIG-17: Graphical Example of Complete Linkage Method

2.2.1.1.3 Average Linkage Method

Here the distance between one cluster and another cluster to be equal to the average distance from any member of one cluster to any member of the other cluster. The dissimilarity between clusters is calculated using average values. The average distance is calculated from the distance between each point in a cluster and all other points in another cluster. The two clusters with the lowest average distance are joined together to form the new cluster. Average-link clustering may cause elongated clusters to split and for portions of neighboring elongated clusters to merge.

$$d_k = \text{sim}(c_i, c_j) = d_{\text{ave}}(c_i, c_j) = (1/n_i n_j) \sum_{x \in c_i} \sum_{y \in c_j} \|x - y\|$$

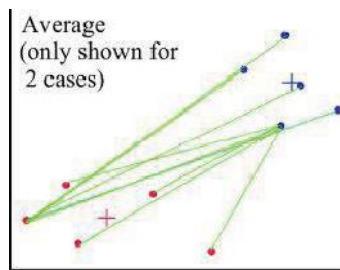


FIG-18: Graphical Example of Average Linkage Method

2.2.1.2 Divisive Analysis (DIANA)

Clusters are divided until cluster with the largest diameter has the largest diameter value smaller than a prescribed threshold or each cluster contains only a single observation or a requested number K of clusters is achieved.

○ DIANA:

- 1) Initially, n genes in input dataset form a single cluster.
- 2) Find the gene g_i , which has the highest average dissimilarity to all other genes. This gene g_i from cluster C initiates a new cluster CNEW.
 - For each gene g_j outside the new cluster CNEW compute D_i .

$$D_i = (\text{average } D(g_i, g_j), g_j \notin \text{CNEW}) - (\text{average } D(g_i, g_j), g_j \in \text{CNEW})$$

- Find a gene g_h for which the difference D_h is the largest. If D_h is positive, then g_h is, on the average close to the new cluster CNEW. Assign gene g_h to new cluster CNEW.
- Repeat Steps until all differences D_h are negative.

- 3) Repeat steps until the highest average dissimilarity to a gene g_i with all other genes become greater than a threshold distance.

2.2.1.3 Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)

The BIRCH algorithm (Balanced Iterative Reducing and Clustering) is an updated modified version of agglomerative nesting that overcomes the two difficulties of agglomerative clustering methods: (1) scalability and (2) the inability to undo what was done in the previous step. It stores summary information about candidate clusters in a dynamic tree data structure. This tree hierarchically organizes the clusters represented at the leaf nodes. The tree can be rebuilt when a threshold specifying cluster size is updated manually, or when memory constraints force a change in this threshold. This algorithm has a time complexity linear in the number of instances. It is designed for clustering a large amount of numerical data by integration of hierarchical clustering (at the initial micro clustering phase-I) and other clustering methods such as iterative partitioning (at the later macro clustering phase-II). BIRCH introduces two concepts, clustering feature and clustering feature tree (CF tree), which are used to summarize cluster representations.

A clustering feature (CF) is a three-dimensional vector summarizing information about clusters of objects. Given n d -dimensional objects or points in a cluster, $\{x_i\}$, then the CF of the cluster is defined as

$$CF = [n, LS, SS],$$

Where n is the number of points in the cluster, LS is the linear sum of the n points (i.e., $\sum x_i$), and SS is the square sum of the data points (i.e., $\sum x_i^2$).

CF Tree:-

- A non leaf node represents a cluster made up of all sub clusters represented by its entries.
- A leaf node also represents a cluster made up of all sub clusters represented by its entries. The diameter or radius of any entry has to be less than the threshold T .
- The tree size is a function of T . The larger T is the smaller the tree is.
- A CF tree will be built dynamically as new data objects are inserted.
- B and L are determined by page size P .

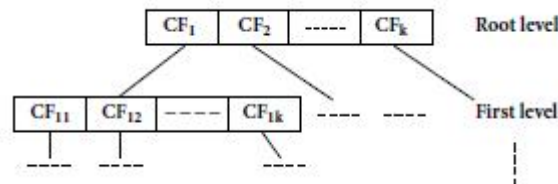


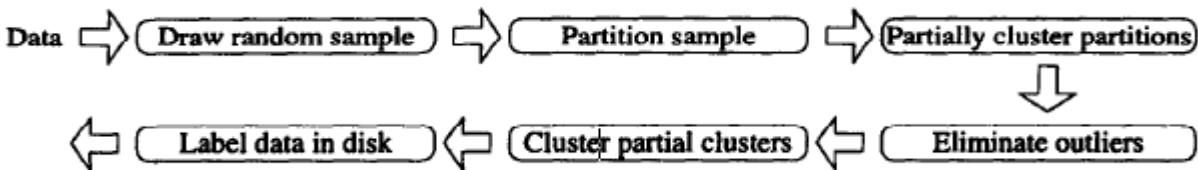
FIG-19: CF Tree

Phases of BIRCH:-

- Use CF (Clustering Feature) tree, a hierarchical data structure for multiphase clustering
 - Phase 1: scan DB to build an initial in-memory CF tree (a multi-level compression of the data into sub-clusters that tries to preserve the inherent clustering structure of the data).
 - Phase 2: use an arbitrary clustering algorithm to cluster the leaf nodes of the CF-tree.

2.2.1.4 Clustering Using Representatives (CURE)

CURE (Clustering Using Representatives) employs a hierarchical clustering algorithm that adopts a middle ground between the centroid-based and the all-point extremes. In CURE, a constant number c of well *scattered* points in a cluster are first chosen. The scattered points capture the shape and extent of the cluster. The chosen scattered points are next shrunk towards the centroid of the cluster by a fraction α . These scattered points after shrinking are used as representatives of the cluster. The clusters with the closest pair of representative points are the clusters that are merged at each step of CURE's hierarchical clustering algorithm.

**FIG-20: Flow of CURE Procedure**CURE Clustering Procedure

1. It is similar to hierarchical clustering approach. But it use sample point variant as the cluster representative rather than every point in the cluster.
2. First set a target sample number c . Then we try to select c well scattered sample points from the cluster.
3. The chosen scattered points are shrunk toward the centroid in a fraction of α where $0 \leq \alpha \leq 1$
4. These points are used as representative of clusters and will be used as the point in d_{\min} cluster merging approach.
5. After each merging, c sample points will be selected from original representative of previous clusters to represent new cluster.
6. Cluster merging will be stopped until target k cluster is found

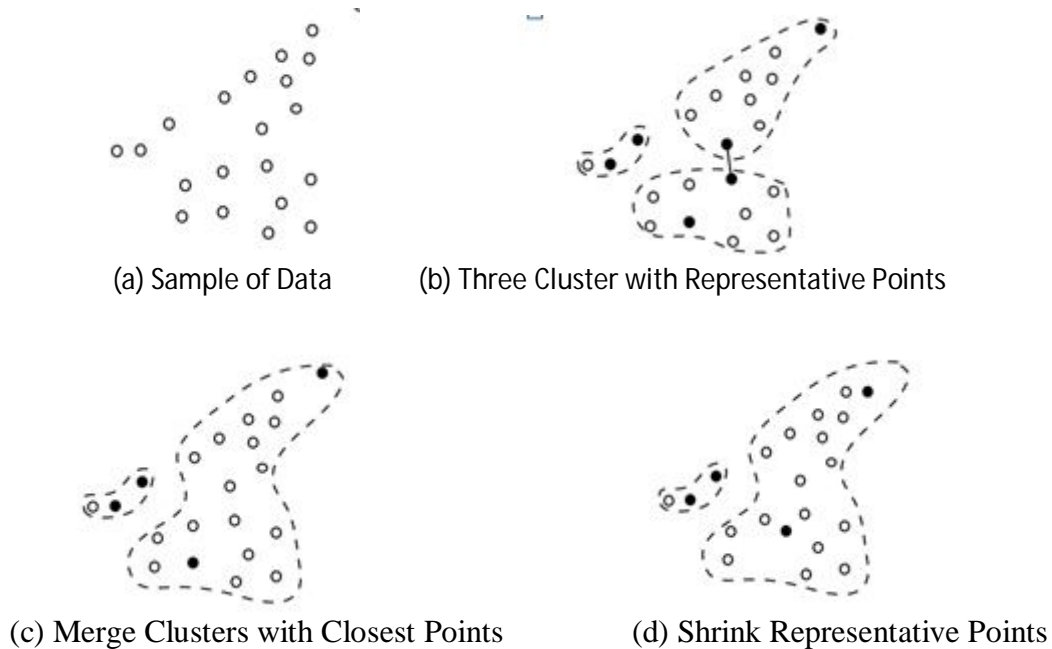


FIG-21: CURE Approach

2.2.1.4 Robust Clustering using LinKs (ROCK)

ROCK (ROBust Clustering using linKs) is a hierarchical clustering algorithm that uses the concept of links i.e. the number of common neighbors between two objects, for data with categorical attributes. Two points, \mathbf{p}_i and \mathbf{p}_j , are neighbors if $\text{sim}(\mathbf{p}_i, \mathbf{p}_j) \geq \theta$, where sim is a similarity function and θ is a user-specified threshold. We can choose sim to be a distance metric or even a non metric that is normalized so that its values fall between 0 and 1, with larger values indicating that the points are more similar. The number of links between \mathbf{p}_i and \mathbf{p}_j is defined as the number of common neighbors between \mathbf{p}_i and \mathbf{p}_j . If the number of links between two points is large, then it is more likely that they belong to the same cluster. By considering neighboring data points in the relationship between individual pairs of points, ROCK is more robust than standard clustering methods that focus only on point similarity.

2.2.1.5 CHAMELEON

CHAMELEON is a hierarchical clustering algorithm that uses dynamic modeling to determine the similarity between pairs of clusters. CHAMELEON operates on a sparse graph in which nodes represent data items, and weighted edges represent similarities among the data items. This sparse graph representation of the data set allows CHAMELEON to scale to large data sets. CHAMELEON finds the clusters in the data set by using a two phase algorithm. During the first phase, CHAMELEON uses a graph partitioning algorithm to cluster the data items into a large number of relatively small sub-clusters. During the second phase, it uses an agglomerative hierarchical clustering algorithm to find the genuine clusters by repeatedly combining together these sub-clusters.

The graph-partitioning algorithm partitions the k-nearest-neighbor graph such that it minimizes the edge cut. That is, a cluster C is partitioned into sub clusters C_i and C_j so as to minimize the weight of the edges that would be cut should C be bisected into C_i and C_j . Edge cut is denoted $EC(C_i, C_j)$ and assesses the absolute interconnectivity between clusters C_i and C_j . Chameleon determines the similarity between each pair of clusters C_i and C_j according to their relative interconnectivity, $RI(C_i, C_j)$,

$$RI(C_i, C_j) = \frac{|EC_{\{C_i, C_j\}}|}{\frac{1}{2}(|EC_{C_i}| + |EC_{C_j}|)}$$

$EC_{\{C_i, C_j\}}$ is the edge cut, defined as minimum sum of the cut edges that would be cut from a cluster C to bisect it into C_i and C_j . EC_{C_i} is the minimum sum of the cut edges that partition C_i into two roughly equal parts.

Their relative closeness,

$$RC(C_i, C_j) = \frac{\bar{S}_{EC_{\{C_i, C_j\}}}}{\frac{|C_i|}{|C_i|+|C_j|} \bar{S}_{EC_{C_i}} + \frac{|C_j|}{|C_i|+|C_j|} \bar{S}_{EC_{C_j}}}$$

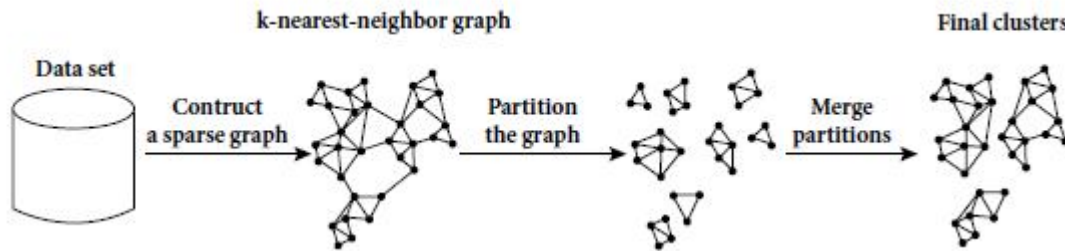


FIG-22: CHAMELEON

2.2.2 Partitional Clustering

Partitional Clustering relocate instances by moving them from one cluster to another, starting from an initial partitioning. Such methods typically require that the number of clusters will be pre-set by the user. A partitional clustering algorithm obtains a single partition of the data instead of a clustering structure. They take the number of cluster K as input. They try to discover clusters by iteratively relocating points between subsets. Certain heuristics are used in the form of iterative optimization. If D is a data set of n objects, and k , the number of clusters to form, a partitioning algorithm organizes the objects into k partitions ($k \leq n$), where each partition represents a cluster. Partitional Algorithms includes K-means, EM clustering, PAM, CLARA and CLARANS.

2.2.2.1 K-means

This method initially takes the number of components of the population equal to the final required number of clusters. In this step itself the final required number of clusters is chosen such that the points are mutually farthest apart. Next, it examines each component in the population and assigns it to one of the clusters depending on the minimum distance. The centroids position is recalculated every time a component is added to the cluster and this continues until all the components are grouped into the final required number of clusters.

Suppose we have n feature vectors X_1, X_2, \dots, X_n all belonging to the same class C and we know that they belong to k clusters such that $k < n$. If clusters are well separated we can use a minimum distance classifier to separate them. We first initialize the means $\mu_1 \dots \mu_k$ of k clusters. One of the ways to do this is just to assign random numbers to them. We then determine the membership of each X by taking the $\| X - \mu_i \|$. The minimum distance determines X 's membership in a respective cluster.

Algorithm: k-means

The k -means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

Input:

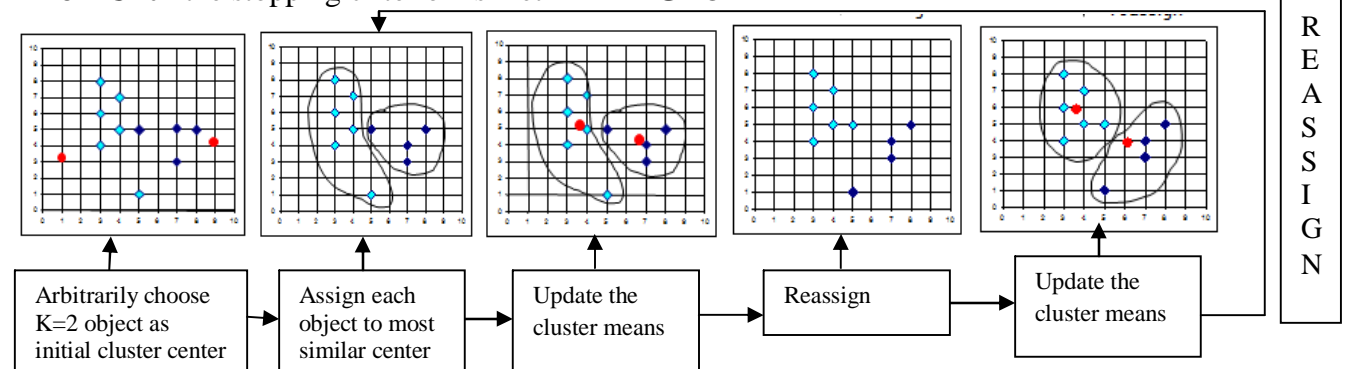
- k : the number of clusters,
- D : a data set containing n objects.

Output: A set of k clusters.

Method:

- 1 Choose k data points as the initial centroids (cluster centers)
- 2 **Repeat**
- 3 **For** each data point $x \in D$ do
- 4 Compute the distance from x to each centroid;
- 5 assign x to the closest centroid // a centroid represents a cluster
- 6 **End For**
- 7 Re-compute the centroids using the current cluster memberships
- 8 **Until** the stopping criterion is met

FIG-23



2.2.2.2 **ISODATA**

The ISODATA method is a method which added division of a cluster, and processing of fusion to the K-means method. The individual density of a cluster is controllable by performing division and fusion to the cluster generated from the K-means method. The individual in a cluster divides past [a detached building] and its cluster, and the distance between clusters unites them with past close. The parameter which set up division and fusion beforehand determines. The procedure of the ISODATA method is shown as follows.

- 1) Parameters, such as the number of the last clusters, a convergence condition of rearrangement, judgment conditions of a minute cluster, branch condition of division and fusion, and end conditions, are determined.
- 2) The initial cluster center of gravity is selected.
- 3) Based on the convergence condition of rearrangement, an individual is rearranged in the way of the K-means method.
- 4) It considers with a minute cluster that it is below threshold with the number of individuals of a cluster, and excepts from future clustering.
- 5) When it is more than the threshold that exists within fixed limits which the number of clusters centers on the number of the last clusters, and has the minimum of the distance between the cluster center of gravity and is below threshold with the maximum of distribution in a cluster, clustering regards it as convergence and ends processing. When not converging, it progresses to the following step.
- 6) If the number of clusters exceeds the fixed range, when large, a cluster is divided, and when small, it will unite. It divides, if the number of times of a repetition is odd when there is the number of clusters within fixed limits, and if the number is even, it unites. If division and fusion finish, it will return to 3 and processing will be repeated.
- 7) Division of a cluster: If it is more than threshold with distribution of a cluster, carry out the cluster along with the 1st principal component for 2 minutes, and search for the new cluster center of gravity. Distribution of a cluster is re-calculated, and division is continued until it becomes below threshold.
- 8) Fusion of a cluster: If it is below threshold with the minimum of the distance between the cluster centers of gravity, unite the cluster pair and search for the new cluster center of gravity. The distance between the cluster center of gravity is re-calculated, and fusion is continued until the minimum becomes more than threshold.

2.2.2.3 Partitioning Around Medoids (PAM)

One of the important partitioning algorithms, which attempt to minimize the Sum of Square Error, is the K -medoids or PAM (Partition Around Medoids—(Kaufmann and Rousseeuw, 1987)). This algorithm is very similar to the K -means algorithm. It differs from the latter mainly in its representation of the different clusters. Each cluster is represented by the most centric object in the cluster, rather than by the implicit mean that may not belong to the cluster. The K -medoids method is more robust than the K -means algorithm in the presence of noise and outliers because a medoid is less influenced by outliers or other extreme values than a mean. However, its processing is more costly than the K -means method. Both methods require the user to specify K , the number of clusters.

Algorithm: k -medoids.

Input:

k : the number of clusters,

D : a data set containing n objects.

Output: A set of k clusters.

Method:

- (1) Arbitrarily choose k objects in D as the initial representative objects or seeds;
- (2) Repeat
- (3) Assign each remaining object to the cluster with the nearest representative object;
- (4) Randomly select a non representative object, $\mathbf{o}_{\text{random}}$;
- (5) Compute the total cost, S , of swapping representative object, \mathbf{o}_j , with $\mathbf{o}_{\text{random}}$;
- (6) If $S < 0$ then swap \mathbf{o}_j with $\mathbf{o}_{\text{random}}$ to form the new set of k representative objects;
- (7) Until no change;

2.2.2.4 CLARANS

CLARANS (A Clustering Algorithm based on Randomized Search) is the clustering process can be presented as searching a graph where every node is a potential solution that is a set of K medoids. Instead of taking the whole set of data into consideration, a small portion of the actual data is chosen as a representative of the data. Medoids are then chosen from this sample using PAM. If the sample is selected in a fairly random manner, it should closely represent the original data set. The representative objects (medoids) chosen will likely be similar to those that would have been chosen from the whole data set. CLARA draws multiple samples of the data set, applies PAM on each sample, and returns its best clustering as the output. CLARA can deal with larger data sets than PAM.

- a) Starting with an arbitrary node (a set of K medoids) randomly check max neighbor (e.g., 100) neighbors.
- b) If a neighbor represents a better partition, the process continues with this new node.
- c) Otherwise a local minimum is found, and the algorithm restarts until numlocal local minima are found (value numlocal=2 is recommended).
- d) The best node (set of medoids) is returned for the formation of a resulting partition.

2.2.2.5 **K-mode**

The K-modes algorithm extends K-means paradigm to cluster categorical data by removing the limitation imposed by K-means through following modifications:

- Using a simple matching dissimilarity measure or the hamming distance for categorical data objects.
- Replacing means of clusters by their modes.

Algorithm: k-mode

- a) Select K initial modes, one for each of the cluster.
- b) Allocate data object to the cluster whose mode is nearest to it according to below equn.

$$d(X, Y) = \sum_{j=1}^F \delta(x_j, y_j)$$

$$\text{where } \delta(x_j, y_j) = \begin{cases} 0 & (x_j = y_j) \\ 1 & (x_j \neq y_j) \end{cases}$$

- c) Compute new modes of all clusters.
- d) Repeat step b) to c) until no data object has changed cluster membership.

2.2.3 **Distance Based Clustering**

An important component of a clustering algorithm is the distance measure between data points. If the components of the data instance vectors are all in the same physical units then it is possible that the simple Euclidean distance metric is sufficient to successfully group similar data instances. Generally, the distance between two points is taken as a common metric to assess the similarity among the components of a population. The commonly used distance measure is the Euclidean metric which defines the distance between two points $p = (p_1, p_2, \dots)$ and $q = (q_1, q_2, \dots)$ is given by :

$$d = \sqrt{\sum_{i=1}^k (p_i - q_i)^2}$$

For higher dimensional data, a popular measure is the Murkowski metric,

$$d_p(x_i, x_j) = \left(\sum_{k=1}^d \|x_{i,k} - x_{j,k}\|^p \right)^{1/p}$$

Where d is the dimensionality of the data. The *Euclidean* distance is a special case where $p=2$, while *Manhattan* metric has $p=1$. There are no general theoretical guidelines for selecting a measure for any given application.

Procedure

- Assign a distance measure between data
- Find a partition such that:
 - Distance between objects within partition (i.e. same cluster) is minimized.
 - Distance between objects from different clusters is maximised.

2.2.3.1 Nearest Neighbour Clustering

Nearest Neighbour clustering algorithm consists of two stages. At the first stage, the k th nearest neighbour density estimation procedure is used to obtain a uniformly consistent estimate of the underlying density. The tree of sample high-density clusters defined on the estimated density is computed at the second stage of the algorithm. At this latter stage, a distance matrix is first computed in which the distance between two "Neighbouring" points (i.e. points with the property that at least one point is one of the k th nearest neighbour of the other) is defined to be inversely proportional to a pooled density estimate at the point halfway between them, and the single linkage clustering algorithm is then applied to this distance matrix to obtain the tree of sample clusters.

2.2.4 Fuzzy Clustering

Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters. One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) Algorithm (Bezdek 1981). In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster. In fuzzy clustering (also referred to as soft clustering), data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster. Fuzzy clustering is a process of assigning these membership levels, and then using them to assign data elements to one or more clusters.

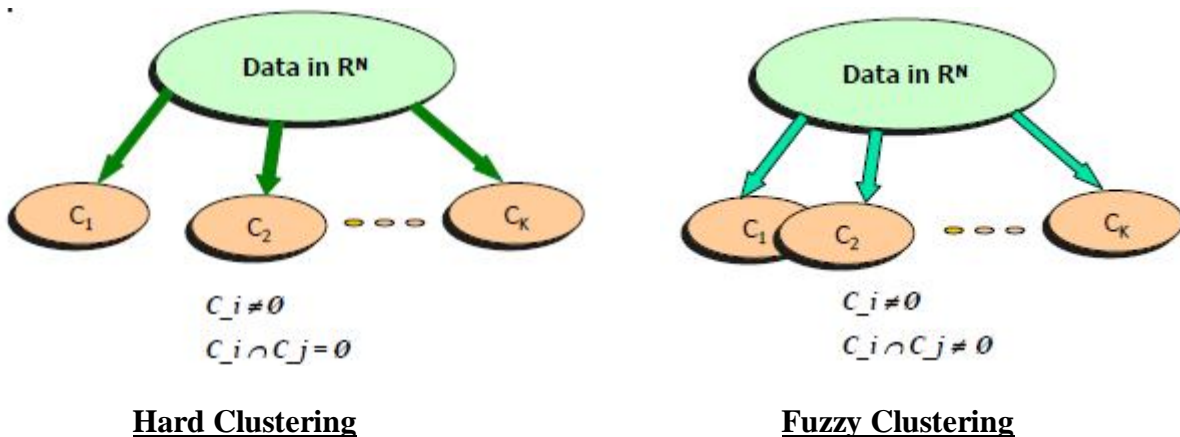


FIG-24

2.2.4.1 Fuzzy c-means (FCM)

The FCM algorithm attempts to partition a finite collection of n elements $X = \{x_1, x_2, \dots, x_n\}$ into a collection of c fuzzy clusters with respect to some given criterion. Given a finite set of data, the algorithm returns a list of c cluster centers $C = \{c_1, \dots, c_c\}$ and a partition matrix $W = w_{ij} \in [0, 1]$, $i=1, \dots, n$, $j=1, \dots, c$. where each element w_{ij} tells the degree to which element x_j belongs to cluster c_j . Like the K-means clustering, the FCM aims to minimize an objective function:

$$\arg \min_C \sum_{i=1}^n \sum_{j=1}^c w_{ij}^m \|x_i - c_j\|^2$$

Where m is any real number greater than 1, u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i th of d -dimensional measured data, c_j is the d -dimension center of the cluster. Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the cluster centers c_j by:

$$w_{ij}^m = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}, \quad c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m}$$

2.2.4.2 Gustafson Kessel (GK)

GK clustering algorithm gives the clusters with the different sizes in the different dimensions and clusters are ellipsoidal in shape. The GK clustering algorithm uses an adaptive distance norm in order to detect the clusters with the different sizes in the different dimensions. This method iteratively improves a sequence of sets of clusters until no further improvement in objective function is possible. The main difference between GK and FCM is in the equation of u_{ij} shown

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{ik}} \right)^{\frac{2}{m-1}}}$$

The distance d_{ij} used by the GK algorithm is

$$d_{ij}^2 = (x_i - \bar{x}_j) A_j (x_i - \bar{x}_j)^T$$

2.2.5 Evolutionary Clustering

Evolutionary clustering is the problem of processing times- tamped data to produce a sequence of clustering; that is, a clustering for each time step of the system. Each clustering in the sequence should be similar to the clustering at the previous time step, and should accurately reflect the data arriving during that time step. An evolutionary clustering should simultaneously

optimize two potentially conflicting criteria: first, the clustering at any point in time should remain faithful to the current data as much as possible; and second, the clustering should not shift dramatically from one time step to the next. Evolutionary clustering refers to the application of evolutionary algorithms (also known as genetic algorithms) to data clustering (or cluster analysis), a general class of problems in machine learning, with numerous applications throughout science and industry.

2.2.5.1 GA Based Clustering

Representation:

Cluster centers encoded in the chromosomes

For a d-dimensional space

Length of chromosome = $d * K$

Population Initialization:

Initial cluster centers = c randomly selected points from the data

For each chromosome i in the population

For each cluster j

P = randomly chosen points from data set

Population[i][j]=p

End

End

Fitness Computation:

Phase 1- Cluster Assignment

Each point is assigned to the nearest cluster center

Phase 2- The cluster centers encoded in the chromosome are replaced by the mean points of the respective clusters.

Phase 3- Fitness = (1/Clustering metric J)

$$J = \sum \sum d^2(x_k^j, v_j), j=1, \dots, c$$

Crossover:

Single point crossover with a fixed crossover probability.

Mutation:

A number δ in the range [0, 1] is generated with uniform distribution. If the value at a gene position is v, after mutation it becomes

$$v = v + 2 * \delta * v \text{ if } v < 0$$

$$v = v + 2 * \delta \text{ if } v = 0$$

Termination:

GA clustering is run for fixed number of iteration. Elitism is incorporated. Best string (Lowest J) is taken as solution of the clustering.

2.2.5.2 VGA Clustering

Each chromosome encodes different number of clusters.

Representation:

For a d-dimensional space, chromosome length = $d \cdot c_i$

$c_i = \text{rand}() \bmod k^* + 2$, k^* is the soft estimation of the upper bound of number of cluster.

Fitness Computation:

Phase 1 - Each point is assigned to the nearest cluster center.

Phase 2 - Compute new centroids and replace them in the chromosome.

Phase 3 - Use cluster validity index as fitness criterion

Crossover:

Parent chromosome P_1, P_2 have C_1, C_2 cluster centers respectively

C_1 , crossover point in $P_1 = \text{rand}() \bmod C_1$

C_2 , crossover point in $P_2 = \text{LB} \leq C_2 \leq \text{UB}$

$\text{LB}(C_2) = \min[2, \max(0, 2 - (C_1 - C_1))]$

$\text{UB}(C_2) = \max[C_2 - \max(0, 2 - C_1)]$

$C_2 = \text{LB}(C_2) + \text{rand}() \bmod [\text{UB}(C_2) - \text{LB}(C_2)]$

Mutation:

A number δ in the range $[0, 1]$ is generated with uniform distribution. If the value at a gene position is v , after mutation it becomes

$$v = v + 2 \cdot \delta \cdot v \quad \text{if } v < 0$$

$$v = \pm 2 \cdot \delta \quad \text{if } v = 0$$

2.2.6 Model-Based Clustering

Model-based clustering methods attempt to optimize the fit between the given data and some mathematical model. Such methods are often based on the assumption that the data are generated by a mixture of underlying probability distributions.

2.2.6.1 Expectation Maximization

- o Make an initial guess of the parameter vector: This involves randomly selecting k objects to represent the cluster means or centers (as in k -means partitioning), as well as making guesses for the additional parameters.
- o Iteratively refine the parameters (or clusters) based on the following two steps:
 - Expectation step: assign points to clusters

$$P(d_i \in c_k) = w_k \Pr(d_i | c_k) / \sum_j w_j \Pr(d_i | c_j)$$

$$w_k = \frac{\sum_i \Pr(d_i \in c_k)}{N}$$

- Maximization step: estimate model parameters

$$\mu_k = \frac{1}{m} \sum_{i=1}^m \frac{d_i P(d_i \in c_k)}{\sum_k P(d_i \in c_j)}$$

2.2.6.2 Artificial Neural Network Based Clustering

Clustering is an unsupervised classification technique that identifies some inherent structures present in a data set based on a similarity or proximity measure. Since all the classification procedures look for an accurate function that underlies the functional relationship between various groups present in the data, artificial neural network proves to be a good option as it can approximate any function with arbitrary accuracy[8,9]. Some of the features of the ANNs that are important in pattern clustering are:

- (1) ANNs process numerical vectors and so require patterns to be represented using quantitative features only.
- (2) ANNs are inherently parallel and distributed processing architectures.
- (3) ANNs may learn their interconnection weights adaptively [Jain and Mao 1996; Oja 1982]. More specifically, they can act as pattern formalizers and feature selectors by appropriate selection of weights.

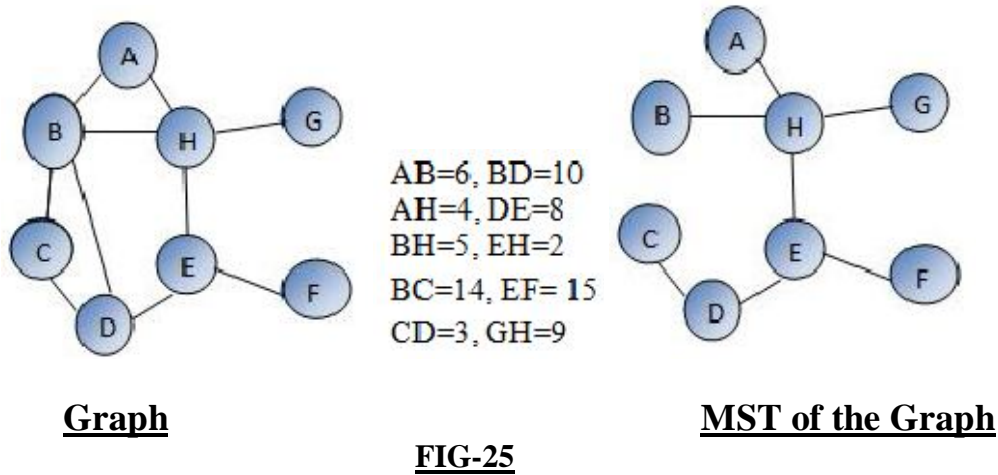
Neural networks have proved to be a useful technique for implementing competitive learning based clustering, which have simple architectures. Such networks have an output layer termed as the competition layer. The neurons in the competition layer are fully connected to the input nodes. The lateral connections in this layer are used to perform lateral inhibition. The basic principle underlying competitive learning is the mathematical statistics problem called cluster analysis that is usually based on the minimization of the average of the squared Euclidean distances between the inputs and their closest prototypes such that the input only attracts its winning prototype and has no effect on the non winning prototypes. Patterns are presented at the input and are associated with the output nodes. The weights between the input nodes and the output nodes are iteratively changed until a termination criterion is satisfied. They are sometimes referred to as Kohonen self-organizing feature maps (SOM), after their creator, Teuvo Kohonen, or as topologically ordered maps. SOMs' goal is to represent all points in a high-dimensional source space by points in a low-dimensional (usually 2-D or 3-D) target space, such that the distance and proximity relationships (hence the topology) are preserved as much as possible. The method is particularly useful when a nonlinear mapping is inherent in the problem itself.

2.2.7 Graph-Based Clustering

Graph based clustering is one of the important type of clustering which is based on the graph theory and has a wide range of application social network analysis, gene expression data analysis, financial market analysis etc. It is very useful on those clustering where information is organized in large data sets, because in that case representation of information using graph is very easy to handle, have faster access and also for clustering. Here objects are represented as nodes in a graph. An edge represents the connection between two nodes. Each edge weight is defined by the Euclidian distance between the two nodes. Mainly Minimum Spanning Tree (MST) is extracted from the graph that is used for clustering which is called as MST based clustering. The MST based clustering is discussed below. Due to their ability to detect clusters with irregular boundaries, MST based clustering algorithms have been widely used in practice.

2.2.7.1 Minimum Spanning Tree (MST) Based Clustering

Minimum Spanning Tree (MST) of a graph is a tree which has all the vertices of the graph as nodes such that the total edge weight of the tree is minimum. Let $G = (V, E)$ be a connected, undirected graph. For each edge (u, v) in E , we have a weight $w(u, v)$ specifying the cost (length of edge) to connect u and v . An acyclic (no circuit) subset T of E that connects all of the vertices in V and whose total weight is minimized, such a tree is called Minimum Spanning Tree (MST). MST can be constructed using any of the two algorithms- Kruskal's algorithm or Prim's algorithm.



The main procedure of MST based clustering algorithm is discussed below.

- a) We have to construct the MST from the graph.
- b) After constructing of MST from the graph we have to detect longer inconsistent edges belongs to the MST. A simple inconsistency measure is the Euclidian distance between two nodes.
- c) After identifying those inconsistent longer edges we have to remove those edges from the MST. After removing $n-1$ inconsistent edges n number of clusters will be formed.

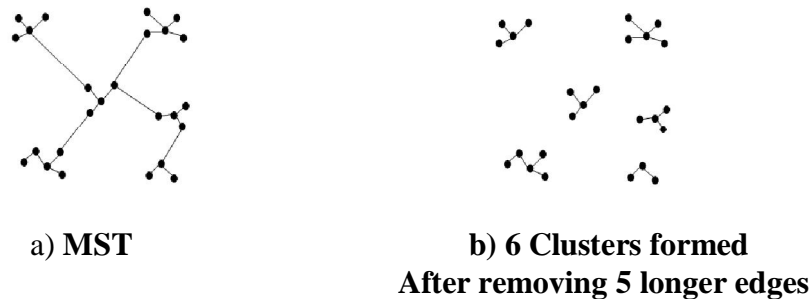


FIG-26: MST Based Clustering

2.2.8 Density-Based Clustering

Density-based Clustering locates regions of high density that are separated from one another by regions of low density. In density-based clustering, [89] clusters are defined as areas of higher density than the remainder of the data set. Objects in these sparse areas - that are required to separate clusters are usually considered to be noise and border points. The main objective of density based clustering was to discover the clusters having arbitrary shape and arbitrary density. The most popular type of density based clustering algorithm includes DBSCAN, DENCLUE and OPTICS.

2.2.8.1 DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density based clustering algorithm that grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases with noise. DBSCAN finds all clusters properly, independent of the size, shape, and location of clusters to each other. DBSCAN is based on two main concepts: density reachability and density connectability. These both concepts depend on two input parameters of the DBSCAN clustering: the size of epsilon neighbourhood ϵ and the minimum points in a cluster *MinPts*. In the clustering process, DBSCAN starts with an arbitrary point p and retrieves all points density-reachable from point p with respect to a given *eps* and *MinPts*. The said two parameters are given arbitrary values initially.

Two global parameters of DBSCAN algorithms are:

- *eps*: Maximum radius of the neighborhood.
- *MinPts*: Minimum number of points in an Eps neighborhood of that point.

Directly Density-Reachable: A point p is directly density-reachable from a point q with respect to *eps*, *MinPts* If p belongs to $NEps(q)$
 $|NEps(q)| \geq MinPts$.

Density-Reachable: A point p is density-reachable from a point q with respect to *eps*, *MinPts* if there is a chain of points $\{p_1 \dots p_n\}$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i .

Density-Connected: A point p is density-connected to a point q with respect to *eps*, *MinPts* if there is a point 'o' such that both, p and q are density-reachable from 'o' with respect to *eps* and *MinPts*.

The procedure of DBSCAN is discussed in below.

- a. Arbitrary selection of a point p
- b. Retrieve all points density-reachable from p with respect to *eps* and *MinPts*.
- c. If p is a core point, then a cluster is formed.
- d. If p is a border point, no points are density reachable from p and DBSCAN visits the next point of the database.
- e. Continue the process until all of the points have been processed.

2.2.8.2 **OPTICS**

Due to the difficultness of selection of parameter values in DBSCAN algorithm in real world high dimensional data set made it difficult to get clustering solution. OPTICS (Ordering Points to Identify the Clustering Structure) can easily overcome this difficulty. OPTICS builds an augmented ordering of data and stores two additional fields with each point, the core-distance and reachability-distance. The core-distance of an object x is the smallest ϵ' value that makes x a core object. The reachability-distance of an object y with respect to another object x is the greater value of the core-distance of x and the Euclidian distance between x and y . In OPTICS this order selects an object that is density reachable with respect to the lowest value so that clusters with higher density (lower ϵ) will be finished first.

2.2.8.3 **DENCLUE**

DENCLUE (DENSity-based CLUstEring) is a clustering method based on a set of density distribution functions. The basic steps included in this procedure are discussed below.

- a. Derive a density function for the space occupied by the data points.
- b. Identify the points that are local maxima. (These are the density attractors.)
- c. Associate each point with a density attractor by moving in the direction of maximum increase in density.
- d. Define clusters consisting of points associated with a particular density attractor.
- e. Discard clusters whose density attractor has a density less than a user-specified threshold of ξ .
- f. Combine clusters that are connected by a path of points that all have a density of ξ or higher.

Density Attractor

A point x^* is called a density attractor for a given influence function, iff x^* is a local maximum of the density-function. Density-attractor are determined by a gradient-based hill-climbing method.

2.2.9 **Grid-Based Clustering**

The grid-based clustering approach uses a multi resolution grid data structure. It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. It has fast processing time, which is typically independent of the number of data objects, Works well for large dimensional data, yet dependent on only the number of cells in each dimension in the quantized space. This type of clustering includes STING and Wave Cluster.

2.2.9.1 **STING**

STING (STatistical INformation Grid) is a grid based clustering where spatial area is divided into rectangular cells and several levels of cells corresponding to different levels of resolution.

Each cell at a high level is partitioned into a number of smaller cells in the next lower level. Statistical info of each cell is calculated and stored beforehand and is used to answer queries. Parameters of higher level cells can be easily calculated from parameters of lower level cell. All Measures are accumulated starting from bottom level cells, and further propagate to higher level cells (e.g., minimum is equal to a minimum among the children-minimums). For each cell in the current layer it is determined that if the cell is relevant at some confidence level. Processing of the next lower level examines only the remaining relevant cells. After all cells are labeled as relevant or not relevant, all regions/clusters that satisfy the density specified can easily be found by Breadth First Search. For a relevant cell, we examine cells within a certain distance d (usually set to side length of bottom layer cell) from the center of the current cell to see if the average density within this small area is greater than density specified. If yes the cells are put into a queue, Repeat steps for all the cells in the queue except cells previously examined are omitted. When the queue is empty we get one region.

2.2.9.2 Wave Cluster

Wave Cluster is basically clustering using wavelet transformation to transform the original feature space that stores the data in a multidimensional grid structure. Then it finds the dense region in transformed space.

2.2.10 Subspace Clustering

The limitation in most of the clustering is the clustering of high dimensional data set. Subspace clustering overcomes this challenge. When dimensionality increases, data become increasingly sparse because the data points are likely located in different dimensional subspaces. When the data become really sparse, data points located at different dimensions can be considered as all equally distanced, and the distance measure, which is essential for cluster analysis, becomes meaningless. To overcome this difficulty, we may consider using feature (or attribute) transformation and feature (or attribute) selection techniques. This type of clustering use the concept of density-based clustering combined with the concept of grid-based clustering. It is based on the observation that different subspaces may contain different, meaningful clusters. Subspace clustering searches for groups of clusters within different subspaces of the same data set. Subspace clustering includes CLIQUE, PROCLUS a dimension-reduction projected clustering and CLICK.

2.2.10.1 CLICK

CLICK (CLuster Identification via Connectivity Kernels) seeks to identify highly connected components in the proximity graph as clusters. The weight, w_{ij} of an edge (i, j) in the proximity graph is defined as the probability that vertices i and j are in the same cluster. The clustering process of CLICK iteratively finds the minimum cut in the proximity graph and recursively splits the data set into a set of connected components from the minimum cut.

2.2.10.2 **CLIQUE**

CLIQUE (Clustering In QUEst) is fundamental to subspace clustering. CLIQUE clustering identifies the sparse and the dense areas in space (or units) in a large set of multi dimensional data points where the data space is not uniformly occupied by the data points. The basic steps included in this clustering are defined in below.

- Find all the dense areas in the one-dimensional spaces corresponding to each attribute. This is the set of dense one-dimensional cells.
- Set $k = 2$.
 - a. Generate all candidate dense k -dimensional cells from dense $(k - 1)$ -dimensional cells.
 - b. Eliminate cells that have fewer than points.
 - c. Set $k = k + 1$.
- Repeat steps until there are no candidate dense k -dimensional cells.
- Find clusters by taking the union of all adjacent, high-density cells.
- Summarize each cluster using a small set of inequalities that describe the attribute ranges of the cells in the cluster.

Chapter 3:

KOHONEN'S SELF-ORGANIZING MAP

3.1 Introduction

Clustering is an unsupervised classification technique that identifies some inherent structures present in a data set based on a similarity or proximity measure. Since all the classification procedures look for an accurate function that underlies the functional relationship between various groups present in the data, artificial neural network proves to be a good option as it can approximate any function with arbitrary accuracy [11,12]. The important properties those holds by ANN for which it becomes very popular in clustering are listed below.

- a) Artificial Neural Networks are inherently parallel and distributed processing architectures.
- b) ANN learns by adjusting their interconnection weights so as to best fit the data. This allows them to “normalize” or “prototype the patterns and act as feature (or attribute) extractors for the various clusters.
- c) ANN process numerical vectors and require object patterns to be represented by quantitative features only. Many clustering tasks handle only numerical data or can transform their data into quantitative features if needed.

Neural networks have proved to be a useful technique for implementing competitive learning based clustering, which have simple architectures. Such networks have an output layer termed as the competition layer. The neurons in the competition layer are fully connected to the input nodes. The lateral connections in this layer are used to perform lateral inhibition. The basic principle underlying competitive learning is the mathematical statistics problem called cluster analysis that is usually based on the minimization of the average of the squared Euclidean distances between the inputs and their closest prototypes such that the input only attracts its winning prototype and has no effect on the non winning prototypes. Patterns are presented at the input and are associated with the output nodes. The weights between the input nodes and the output nodes are iteratively changed until a termination criterion is satisfied. A competitive learning-based neural networks used for clustering include Kohonen's Self-organizing map (SOM). SOMs' goal is to represent all points in a high-dimensional source space by points in a low-dimensional (usually 2-D or 3-D) target space, such that the distance and proximity relationships (hence the topology) are preserved as much as possible. The method is particularly useful when a nonlinear mapping is inherent in the problem itself.

The self-organizing map (SOM) is widely applied approach for clustering and pattern recognition that can be used in many stages of the image processing, e. g. in color image segmentation, generation of a global ordering of spectral vectors, image compression, binarisation document etc.

Clustering is an unsupervised method for grouping similar type objects from heterogeneous collection of data points and discover the inherent structure present in the data set. So there is a lot of clustering techniques, but most of the clustering take number of clusters as a input from previous [3] [4] [5]. In that case we need a prior knowledge about number of clusters before starting the clustering. For example; k-means [6] is very popular clustering algorithm, faces same problem in real life. It also inefficient to separate overlapped clusters present in the data set [3]. Also in case of hierarchical agglomerative clustering we have to know where to cut the dendrogram to form the cluster [7]. In the same way, the Density Based Hierarchical (DHC) clustering suffers from computational complexity in case of large data set [8]. As a result in recent era a lot of attention has been paid for creation of such an effective clustering algorithm which can handle the clustering of large or small data set that has arbitrary shape and density, without knowing actual number of cluster present in the data set from previous [9].

Keeping this view in mind we have exploited the potentiality of adaptive and competitive learning process of neural network by using self organizing property of Kohonen Self-Organizing Feature Map (SOM) for our clustering process [12]. SOM is a powerful explanatory tool in case of data analysis and data mining, widely used in pattern recognition, biological modeling, data compression and signal processing [13]. The basic SOM consists of some neurons, usually arranged in a two dimensional structure and the neurons are in a neighborhood relation. After completion of training, each neuron is attached to a feature vector of the same dimension as input space. To form the clusters from whole input space Self-Organizing Map (SOM) assigned the input vector to the neuron with nearest feature vectors. This process can be considered as performing vector quantization (VQ) [14]. We have learned the whole data set using SOM for our clustering process at first and based on positions of output nodes of learned SOM we have assigned the data points into different clusters based on the nearest distances of the data points with output SOM nodes, combining with the help of Minimum Spanning Tree.

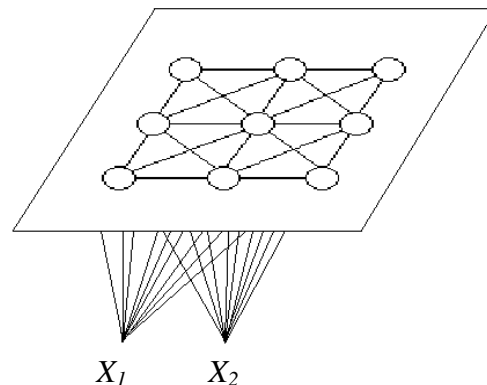


FIG-27: Schematic diagram of Kohonen Self Organizing Map [55]

The SOM is an unsupervised competitive ANN, which transforms highly dimensional data into a two dimensional grid, while keeping the data topology by mapping similar data items to the same cell on the grid (or to neighboring cells). The SOM has a feed-forward structure that consists of two layers; one is computational output layer where single layer of neurons arranged in a two dimensional plane and another layer is the input layer where each of the source nodes is fully connected with neurons of output layer. The number of neurons in the input layer is equal to the dimension of the output layer. Self organization process of SOM is consist of four essential components-

I. Initialization:-

All connection weights between input nodes to output nodes in computation layer are initialized with small random values between [0, 1].

II. Competition:-

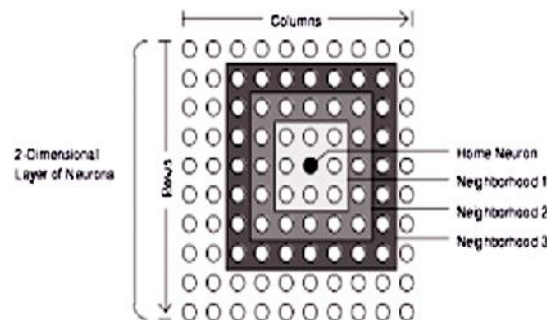
When an input pattern presented to the SOM network, neurons compute their respective values of discriminant function for all weight vectors based on which competition is performed. The discriminant function to be the squared Euclidean distance between the input vector x and the weight vector w_j for each neuron j

$$d_j(x) = \sum_{i=1}^D (x_i - w_{ji})^2$$

The particular neuron that has smallest value of discriminant function in grid matrix is activated, called “winner” neuron in competition (winner take all).

III. Cooperation:-

The winner neuron determines the center of topological neighborhood region of excited neurons, providing the basis for cooperation. The neighborhood area can be expressed as a rectangular field or a hexagonal one. The neighborhood area can be expressed as a rectangular field or a hexagonal one, as showed in the figure below.



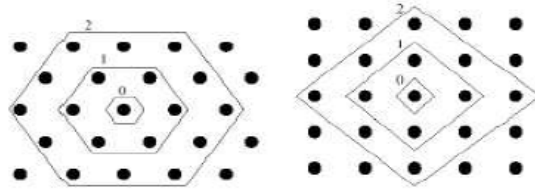


FIG-28: Rectangular and Hexagonal Neighborhood

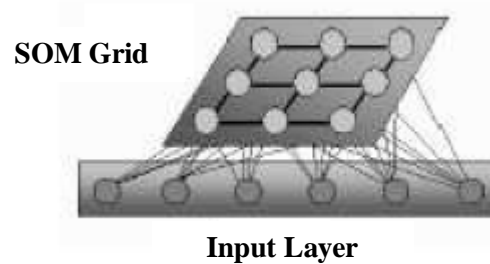


FIG-29: SOM Grid Structure

The lateral interaction between winner and neighbor neurons represented by neighbor topological function whose maximum value is reached when the distance among the winner and neighbor neurons is zero. Otherwise, the minimum function value occurs when the distance among the winner neuron and its neighbors tends to the infinity. The Gaussian function satisfies the above necessities and is widely employed in SOM networks:

$$f_{j,i(x)} = \exp(-d_{j,i}^2/2\sigma^2)$$

IV. Synaptic Adaptation:-

In this phase, the synaptic weights are adjusted so that the winner's weight vector approximates to the input vector x . Once the network is continuously fed with the input set, the algorithm produces a topological map ordination of the features, leading to similar values of the weight vectors for the adjacent neurons. The excited neurons decrease their individual values of the discernment function in relation to the input pattern through suitable adjustment of the associated connection weights.

3.2 Discussion of Self-Organizing Map Algorithm

The SOM Algorithm could be summarized in five steps:

1. Initialization

Attribute random small values to the synaptic weights of each neuron. This step ensures that the map will have no previous organization at all.

2. Sampling

Present a random choose sample X to the network, from the input space with n dimensionality, being

$$X = [x_1, x_2, \dots, x_n]^t$$

3. Matching

Find the winner neuron $i(X)$, with the weight vector, $W = [w_1, w_2, \dots, w_n]^t$ in the epoch t , closer to the vector X presented to the network, adopting the minimum Euclidian distance criterion

$$i(x) = \underset{j}{\operatorname{argmin}} \| X - W_j \|, j = 1, 2, \dots, n$$

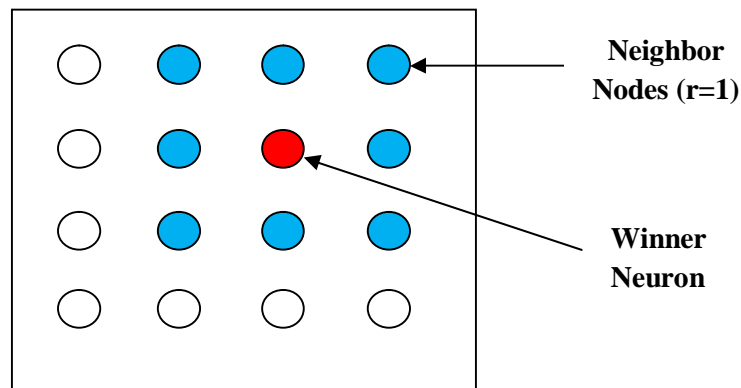


FIG-30

4. Updating

Adjust the synaptic weights to every grid neuron by the actualization formula

$$W_{ij}(t+1) = W_{ij}(t) + \beta(t) * f_{j,i(x)}(t) * (x_i(t) - W_{ij}(t)), [0 < \beta(t) < 1],$$

Where $\beta(t)$ is the learning rate, $f_{j,i(x)}(t)$ is the neighborhood topological function surrounding the winner neuron. Both parameters are dynamically varied to ensure better results.

5. Repetition

Return to step 2 until no significant changes occur in the features map. The competitive learning happens in the second step when the weight vectors are updated. To each input presented to the network only one neuron must be active in a specific instant. In this sense, the competitive learning has a priority to define the statistical features more outstanding for the

classification of an input pattern set [HAYKIN 2001]. In the third step, the cooperative process among the grid neurons is determined by the neighborhood area of the winner unit ($f_j, i(x)$). The network training consists in two distinct phases: the rough phase and the fine tuning phase. The first is characterized by the topological ordination of the weight vectors. In this phase, the learning rate should be set in value close proximity to 0.1 and the neighborhood area of the winner neuron should take almost all neurons of the grid. During the rough phase the learning rate decreases smoothly until it reaches the value of 0.01. The second phase is necessary to achieve a fine-tuning ordination of the features map. To make the statistical precision as good as possible, the learning rate should be maintained closer to 0.01; it should not take zero to avoid a meta-stable state of the grid (a topological impairment). The neighborhood area must contain only the next neighbors of the winner unit, possibly reducing its area to one or zero in the fine-tuning phase.

Training Phase of Self-Organizing Map

A SOM is neural network with unsupervised type of learning, i. e. no cluster values denoting an a priori grouping of the data instances are provided. The learning process is divided in epochs, during which the entire batch of input vectors is processed. The epoch involves the following steps:

1. Consecutive submission of an input data vector to the network.
2. Calculation of a distance between the input vector and the weight vectors of the neurons of the output layer.
3. Selection of the nearest (the most similar) neuron of the output layer to the presented input data vector.
4. An adjustment of the weights.

SOM can be trained in either recursive or batch mode. In recursive mode, the weights of the winning neurons are updated after each submission of an input vector, whereas in batch mode, the weight adjustment for each neuron is made after the entire batch of inputs has been processed, i. e. at the end of an epoch. The weights adapt during the learning process based on a competition, i. e. the nearest (the most similar) neuron of the output layer to the submitted input vector becomes a winner and its weight vector and the weight vectors of its neighbouring neurons are adjusted according to

$$W = W + \lambda * \phi_s (x_i - w)$$

Where W is the weight matrix, x_i the submitted input vector, λ the learning parameter determining the strength of the learning and ϕ_s the neighbourhood strength parameter determining how the weight adjustment decays with distance from the winner neuron (it depends on s , the value of the neighbourhood size parameter). The learning process can be divided into two phases: ordering and convergence. In the ordering phase, the topological ordering of the weight vectors is established using reduction of learning rate and neighbourhood size with

iterations. In the convergence phase, the SOM is fine tuned with the shrunk neighbourhood and constant learning rate.

Parameter Learning

The learning parameter, corresponding to the strength of the learning, is usually reduced during the learning process. It decays from the initial value to the final value, which can be reached already during the learning process, not only at the end of the learning. Some of the important decay functions are in below.

- No decay

$$\lambda_t = \lambda_0,$$

- Linear decay

$$\lambda_t = \lambda_0 \left(1 - \frac{t}{\tau}\right),$$

- Exponential decay

$$\lambda_t = \lambda_0 e^{-\frac{t}{\tau}},$$

Learning of Neighbourhood Neuron

In the learning process not only the winner but also the neighbouring neurons of the winner neuron learn, i.e. adjust their weights. All neighbour weight vectors are shifted towards the submitted input vector, however, the winning neuron update is the most pronounced and the farther away the neighbouring neuron is, the less its weight is updated. This procedure of the weight adjustment produces topology preservation. The initial value of the neighbourhood size can be up to the size of the output layer, the final value of the neighbourhood size must not be less than 1. The neighbourhood strength parameter, determining how the weight adjustment of the neighbouring neurons decays with distance from the winner, is usually reduced during the learning process. It decays from the initial value to the final value, which can be reached already during the learning process, not only at the end of the learning process. The neighbourhood strength parameter should be in the interval [0.01, 1].

Resulting Weight

The resulting weight vectors of the neurons of the output layer, obtained at the end of the learning process, represent the centers of the clusters. The resulting patterns of the weight vectors may depend on the type of the weights initialization.

Distance Measurement

The measure of the distance between the presented input vector and its weight vectors may have many form, some of those are shown below.

- Euclidian Distance

$$d_j = \sqrt{\sum_{i=1}^N (x_i - w_{ji})^2}$$

- Correlation

$$d_j = \sum_{i=1}^N \frac{(x_i - \bar{x})(w_{ji} - \bar{w}_j)}{\sigma_x \sigma_{w_j}}$$

- Direction Cosine

$$d_j = \frac{\sum_{i=1}^N x_i w_{ji}}{\|x_i\| \|w_{ji}\|}$$

Where x_i is i -th component of the input vector, w_{ji} i -th component of the j -th weight vector, N dimension of the input and weight vectors, \bar{x} mean value of the input vector x , \bar{w}_j mean value of the weight vector w_j , σ_x standard deviation of the input vector x , σ_{w_j} standard deviation of the weight vector w_j , $\|x_i\|$ length of the input vector x and $\|w_{ji}\|$ length of the weight vector w_j .

Learning Progress

The learning progress criterion, minimized over the learning process, is the sum of distances between all input vectors and their respective winning neuron weights, calculated after the end of each epoch, according to

$$D = \sum_{i=1}^k \sum_{n \in c_i} (x_n - w_i)^2$$

Where x_n is the n -th input vector belonging to cluster c_i whose center is represented by w_i

Error Calculation

The error of trained SOM can be calculated in many forms shown in below.

- Learning progress criterion as in previous equn.
- Normalized learning progress criterion

$$E = \frac{1}{M} \sum_{i=1}^k \sum_{n \in c_i} (\mathbf{x}_n - \mathbf{w}_i)^2$$

- Normalized error in cluster

$$E = \frac{1}{k} \sum_{i=1}^k \frac{1}{M_i} \sum_{n \in c_i} (\mathbf{x}_n - \mathbf{w}_i)^2$$

- Normalized error in i-th cluster

$$E_i = \frac{1}{M_i} \sum_{n \in c_i} (\mathbf{x}_n - \mathbf{w}_i)^2$$

Where \mathbf{x}_n is the n-th input vector belonging to cluster c_i whose center is represented by \mathbf{w}_i (e. i. the weight vector of the winning neuron representing cluster c_i), M is number of input vectors, M_i is number of input vectors belonging to i-th cluster and k is number of clusters.

Cluster formation

The U-matrix (the matrix of average distance between weights vectors of neighbouring neurons) can be used for finding of realistic and distinct clusters.

Algorithm

Input: X an $(n \times p)$ data set containing p dimensional n number of data points

Outputs: A vector Y , length m : (Y_1, Y_2, \dots, Y_n) and distance matrix z $[p, q]$ between output nodes of SOM

Method:

Step 0 :=> Initialized weights W_{ij} with small random values at time $t=0$

Set topological neighborhood parameters

Set learning rate α

Step 1 :=> $t=t+1$

Step 2 :=> Present new inputs X_i

Step 3 :=> Repeat steps 4-9

Step 4 :=> For each input vector X do step 5-7

Step 5 :=> For each j compute distance d_j between the input and each output node j using

$$d_j = \sum_{i=1}^m (\mathbf{x}_i(t) - \mathbf{W}_{ij}(t))^2 \quad (1)$$

Where $X_i(t)$ is the input to node i at time t and $W_{ij}(t)$ is the weight from input node i to output node j at time t

Step 6 :=> Find index j such that d_j is minimum

Step 7 :=> For all units j with a specified neighborhood of j and for all i

$$W_{ij}(t+1) = W_{ij}(t) + \beta(t) * (x_i(t) - W_{ij}(t)), [0 < \beta(t) < 1] \quad (2)$$

Step 8 :=> Reduce learning rate and radius

$$\alpha(t+1) = \alpha(t) * 0.5$$

Step 9 :=> Until ($\alpha=0$)

Example: A Kohonen self-organizing map (SOM) to cluster four vectors

Let the vector to be clustered be

(1, 1, 0, 0); (0, 0, 0, 1); (1, 0, 0, 0); (0, 0, 1, 1);

The maximum number of clusters to be formed

$$m = 2$$

Suppose the Learning rate (geometric decrease) is

$$\alpha(0) = 0.6$$

$$\alpha(t+1) = 0.5 \times \alpha(t)$$

With only two clusters available, the neighborhood of node J (Step 4) is set so that only one cluster updates its weights at each step (i.e. $R=0$)

Step 0:= Initial weight matrix:

0.2	0.8
0.6	0.4
0.5	0.7
0.9	0.3

Initial Radius:=

$$R = 0$$

$$\alpha(0) = 0.6$$

Step 1:= Begin Training**Step 2:=** For the first vector, (1, 1, 0, 0), do Steps 3-5**Step 3:=** $D(1) = (0.2 - 1)^2 + (0.6 - 1)^2 + (0.5 - 0)^2 + (0.9 - 0)^2 = 1.86$ $D(2) = (0.8 - 1)^2 + (0.4 - 1)^2 + (0.7 - 0)^2 + (0.3 - 0)^2 = 0.98$ **Step 4:=** The input vector is closest to output node 2, so $J=2$ **Step 5:=** The weights on the winning unit are updated:

$$w_{i2}(\text{new}) = w_{i2}(\text{old}) + [x_i - w_{i2}(\text{old})]$$

$$= 0.4 w_{i2}(\text{old}) + 0.6 x_i$$

This gives the weight matrix

$$0.2 \quad 0.92$$

$$0.6 \quad 0.76$$

$$0.5 \quad 0.28$$

$$0.9 \quad 0.12$$

Step 2:= For the second vector, (0, 0, 0, 1), do Steps 3-5**Step 3:=** $D(1) = (0.2 - 0)^2 + (0.6 - 0)^2 + (0.5 - 0)^2 + (0.9 - 1)^2 = 0.66$ $D(2) = (0.8 - 0)^2 + (0.4 - 0)^2 + (0.7 - 0)^2 + (0.3 - 1)^2 = 2.2768$ **Step 4:=** The input vector is closest to output node 1, so $J=1$ **Step 5:=** The weights on the winning unit are updated:

$$w_{i1}(\text{new}) = w_{i1}(\text{old}) + \alpha [x_i - w_{i1}(\text{old})]$$

$$= 0.4 w_{i1}(\text{old}) + 0.6 x_i$$

This gives the weight matrix

$$0.08 \quad 0.92$$

$$0.24 \quad 0.76$$

$$0.20 \quad 0.28$$

$$0.96 \quad 0.12$$

Step 2:= For the third vector, (0, 0, 1, 1), do Steps 3-5**Step 3:=** $D(1) = (0.2 - 0)^2 + (0.6 - 0)^2 + (0.5 - 1)^2 + (0.9 - 1)^2 = 1.8656$ $D(2) = (0.8 - 0)^2 + (0.4 - 0)^2 + (0.7 - 1)^2 + (0.3 - 1)^2 = 0.6768$ **Step 4:=** The input vector is closest to output node 1, so $J=1$ **Step 5:=** The weights on the winning unit are updated:

$$w_{i1}(\text{new}) = w_{i1}(\text{old}) + \alpha [x_i - w_{i1}(\text{old})]$$

$$= 0.4 w_{i1}(\text{old}) + 0.6 x_i$$

This gives the weight matrix

$$\begin{array}{cc} 0.08 & 0.968 \\ 0.24 & 0.304 \\ 0.20 & 0.112 \\ 0.96 & 0.048 \end{array}$$

Step 2:= For the fourth vector, (0, 0, 1, 1), do Steps 3-5

$$\text{Step 3:= } D(1) = (0.2 - 0)^2 + (0.6 - 0)^2 + (0.5 - 1)^2 + (0.9 - 1)^2 = 0.7056$$

$$D(2) = (0.8 - 0)^2 + (0.4 - 0)^2 + (0.7 - 1)^2 + (0.3 - 1)^2 = 2.724$$

Step 4:= The input vector is closest to output node 1, so J=1

Step 5:= The weights on the winning unit are updated:

$$w_{i1}(\text{new}) = w_{i1}(\text{old}) + \alpha [x_i - w_{i1}(\text{old})]$$

$$= 0.4 w_{i1}(\text{old}) + 0.6 x_i$$

This gives another set of weight matrix

Step 6:= reduces the learning rate

$$\alpha = 0.5 \times (0.6) = .3$$

The weight updated equations are now

$$w_{i1}(\text{new}) = w_{i1}(\text{old}) + 0.3 [x_i - w_{i1}(\text{old})]$$

Modifying the adjustment procedure for the learning rate so that it decreases geometrically from 0.6 to .01 over 100 iterations (epochs) gives the following results

Iteration 0	Weight Matrix
	0.2 0.8
	0.6 0.4
	0.5 0.7
	0.9 0.3

Iteration 10 Weight Matrix
1.5e-7 1.0000
4.6e-7 0.3700
0.6300 5.4e-7
1.0000 2.3e-7

Iteration 100 Weight Matrix
6.7e-17 1.0000
2.0e-16 0.4900
0.5100 2.3e-5
1.000 1.3e-16

These weight matrices appear to be converging to the matrix

0.0 1.0
0.0 0.5
0.5 0.0
1.0 0.0

The first column of which is the average of two vectors placed in cluster and the second column of which is the average of the two vectors placed in cluster 2.

Chapter 4: LITERATURE REVIEW

4.1 Self-Organizing Map in Clustering

Self-Organizing Feature Map (SOM) is an unsupervised Artificial Neural Network (ANN) that transforms a high dimensional input space into a low dimensional, typically two dimensional representation of the high dimensional input space. Self-Organizing Map has mainly two layers, input layer and output layer. Output layer consists of neurons arranged in row and column in two dimensional grids, which is fully connected with all the nodes of input layer. It involves four major components in the self organization process- Initialization, Competition, Cooperation and Adaptation. The ordered grid of SOM can be used as a convenient visualization surface for showing different features of the SOM and hence of the data as a cluster structure [25].

In 1992 Lampinen and Oja [27] used a method consisting of a two-level SOM in order to get the hierarchical structure for unknown number of clusters, where the first level SOM is used as a training procedure and second level SOM is used for clustering. Here outputs of the first SOM are fed into a second SOM as inputs. It may be comprises of n independent SOM's organized in layers. The first layer is trained using an input feature vector. Once trained, the winning nodes in the first map act as input to the next layer and so on. The hierarchical map then helps in visualizing the input data at different levels of taxonomic organization. The top layer may be used to identify the presence of clusters and the lower levels representing sub-clusters. This type of SOM can be used to map data sets with higher dimensions. This model was shown to perform better than traditional SOM and classical k-means algorithm in classifying artificial data and sensory information from low-level feature detectors in a computer vision system.

In 1992 Chau-Yun Hsu and Hwai-En Wu [60] proposed an improved algorithm for Kohonen's Self-organizing Feature Maps. About artificial neural network, the well-known architecture and algorithm of self organizing maps has the important property of topology-preserving mapping of various features of input signals and their abstractions with noise involving. However, in the formation (learning) of the mapping, the topological orders preserved in the map are not all "correct", e.g. for some topological orders there is no such feature relations in the input signals; also some "correct" topological orders corresponding to actual feature relations of the input signals are "lost" in the map. So it is more proper to call it a "piece wise-correct" map. This is due to two dominant factors: the initial weight problem and the input sequence order problem. Here we propose a modified algorithm which induces (1) insertion and deletion of cells (2) Coulomb effect of the learning factor to efficiently reduce those two dominant problems and successfully form the topologically correct map. Provided simulation results show the great improvement and excellent performance achieved by the proposed algorithm.

De, A. and Chatterjee, N. have determined the exact nature and location of faults during impulse testing of transformers is of practical importance to the manufacturer as well as designers. The presently available diagnostic techniques more or less depend on expert knowledge of the test personnel, and in many cases are not beyond ambiguity and controversy. They present an artificial neural network (ANN) approach for detection and diagnosis of fault

nature and fault location in oil-filled power transformers during impulse testing. This new approach relies on high discrimination power and excellent generalization ability of ANNs in the complex pattern classification problem, and overcomes the limitations of conventional expert or knowledge-based systems in this field. In the present work, the self-organizing feature map (SOFM) algorithm with Kohonen's learning has been successfully applied to the problem with good diagnostic accuracy.

In 1995 Murtagh et al. [28] and in 2000 Vesanto et al. [29] used hierarchical agglomerative clustering- Single linkage, Average linkage, Complete linkage, Centroid linkage over the output nodes of SOM as a clustering method. The main approach that was used is divided into two-level approach in clustering. First, a large set of prototypes—much larger than the expected number of clusters—is formed using the SOM or some vector quantization algorithm. The prototypes can be interpreted as “proto clusters,” which are in the next step combined to form the actual clusters. Each data vector of the original data set belongs to the same cluster as its nearest prototype. This method reduced the cost of clustering, because it has been already shown that many clustering algorithms, like hierarchical clustering becomes heavy even with small number of samples. For this reason, it is convenient to cluster a set of prototypes rather than directly the data.

In 1995 N Vassila, P Thiran and P Ienne [61] introduce two new variants of Kohonen's self-organizing feature maps based on batch processing are presented in this work. The motivation is related to the need of exploiting the hardware resources of neuron computers based on systolic arrays. Ordering and convergence to asymptotic values for 1-D maps and 1-D continuous input and weight spaces are proved for both variants. Finally, simulations on uniform 2-D data as well as simulations on speech 12-D data using 2-D maps are also presented to back the theoretical results.

In 1996 Kikuo Fujimura et al. [63, 64] introduce the automatic button-color matching system in the textile field using Kohonen's Self-organizing Feature Maps (SOMs). The system consisted of two processes; (a) self-organizing feature mapping and (b) SOM analyzing. The recognition test of the system was performed using an actual data set.

In 1999 Ravi Kothari and Shafiqul Islam [62] presented characterization of soil moisture at a given scale using self-organizing feature maps. It is found that as few as 49 neurons capture the spatial structure of remotely sensed soil moisture images from the southern Great Plains. Average latent heat fluxes computed from the original image of 21204 pixels and from 49 neurons are comparable.

Merkel and Rauber (2000) [32] proposed an extended version of Self-Organizing Map by using capability of growing of map size in high dimensional feature spaces, to uncover the hierarchical structure of text archives. This method overcomes two major challenges for document archive representation-1) the determination of a suitable number of neurons requires some insight into the structure of the document archive. This cannot be assumed, however, in case of unknown document collections. Thus, it might be helpful if the neural network would be able to determine this number during its learning process.2) hierarchical relations between the input data are not mirrored in a straight-forward manner. This neural network architecture is

capable of determining the required number of units during its unsupervised learning process. Additionally, the data set is clustered hierarchically by relying on a layered architecture comprising a number of independent self-organizing maps within each layer. Addition of a single sample with a cluster can radically change the distances [26]. Similar multiple-level approaches to clustering have been proposed earlier as described in [27].

Mangiameli et al. [30], partitive method, a small SOM greatly outperformed hierarchical methods in clustering imperfect data.

In the year **2001 Kiang [31]** proposed more neurons in SOM for training. Here they extend the original SOM network to include a congruity-constrained clustering method to perform clustering based on the output map generated by the network.

In the year **2002 Guterman et al. [33]** for speaker recognition system SOM was used. Here basically a conversation is given and the goals are to estimate the number of participating speakers C and to cluster the conversation into C clusters. Each speaker is modeled by SOM. At the time of starting the training SOMs are randomly initiated. An iterative algorithm allows the data move from one model to another and adjust the SOMs. The restriction that the data can move only in small groups but not by moving each and every feature vector separately forces the SOMs to adjust to speakers (instead of phonemes or other vocal events). This method can be applied to high-quality conversations with two to five participants and to two-speaker telephone-quality conversations.

In the year **2003 Mostafa Allamehzadeh and Mohammad Mokhtari [65]** show that Self- Organizing Feature Maps (SOFM) is powerful intelligent tools and widely in pattern recognition and data clustering. They show that SOFM can be used to predict the concentration and the trend of aftershocks of 1997 Birjand-Ghaen, Iran and 1999 Izmit, Turkey Earthquake.

In the year **2004 Wu et al. [34]** proposed Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density and **Barreto et al. [37]** introduced a vector quantized temporal associative memory (VQTAM) technique of SOM as an alternative of MLP and radial basis function (RBF) neural network for dynamical system identification and control.

In the year **2004 Hamid Moradkhani [66]** shows that Stream flow forecasting which is always been a challenging task for water resources engineers and managers and a major component of water resources system control can be explored with the applicability of a Self Organizing Radial Basis (SORB) function to one-step ahead forecasting of daily stream flow. SORB uses a Gaussian Radial Basis Function architecture in conjunction with the Self-Organizing Feature Map (SOFM) used in data classification. SORB outperforms the two other ANN algorithms, the well known Multi-layer Feed forward Network (MFN) and Self-Organizing Linear Output map (SOLO) neural network for simulation of daily stream flow in the semi-arid Salt River basin. The applicability of the linear regression model was also investigated and concluded that the regression model is not reliable for this study. To generalize the model and derive a robust parameter set, cross-validation is applied and its outcome is compared with the

split sample test. Cross validation justifies the validity of the nonlinear relationship set up between input and output data.

Then in the year **2005 Y.M. Cheung et al. [35]** introduced penalization controlled competitive learning for automatic cluster number selection and clustering of SOM. Already existed Rival Penalized Competitive Learning (RPCL) algorithm and its variants had a ability to perform data clustering without knowing the exact number of clusters. However, their performance is sensitive to the pre selection of the rival de-learning rate. Here they have further investigate the RPCL and present a mechanism to control the strength of rival penalization dynamically.

In the year **2005 AlokMadan [67]** presents a potentially feasible approach for training ANN in active control of earthquake-induced vibrations in building structures without the aid of teacher signals (i.e. target control forces). A counter-propagation neural network is trained to output the control forces that are required to reduce the structural vibrations in the absence of any feedback on the correctness of the output control forces (i.e. without any information on the errors in output activations of the network). The study shows that, in principle, the counter-propagation network (CPN) can learn from the control environment to compute the required control forces without the supervision of a teacher (unsupervised learning). Simulated case studies are presented to demonstrate the feasibility of implementing the unsupervised learning approach in ANN for effective vibration control of structures under the influence of earthquake ground motions. The proposed learning methodology obviates the need for developing a mathematical model of structural dynamics or training a separate neural network to emulate the structural response for implementation in practice.

In the year **2006 Shreyas Sen, Seetharam Narasimhan and Amit Konar [68]** design a scheme for automatic identification of a species from its genome sequence. A set of 64 three tuple keywords is first generated using the four types of bases: A, T, C and G. These keywords are searched on N randomly sampled genome sequences, each of a given length (10,000 elements) and the population count for each of the $4^3 = 64$ keywords is obtained. Clustering techniques are employed on the extracted data from different species using the Self-organizing feature map (SOFM) algorithm. The maps for different dimensions are constructed and analyzed for optimum performance. The scheme presents a novel method for identifying a species from its genome sequence with the help of a two dimensional map of neuronal clusters, where each cluster represents a particular species.

In the year **2008 Sap et al. [39]** proposed a two level clustering algorithm based on SOM and Rough Set theory for detection of overlapping cluster. Most of the time it has been shown that clusters has no crisp boundaries to differentiate one cluster from another smoothly. To overcome these uncertainties this clustering algorithm has proposed by Sap et al. where a rough set incremental clustering of SOM is used. Incremental clustering [3] is based on the assumption that, it is possible to consider data points one at a time and assign them to existing clusters. Thus, a new data item is assigned to a cluster without looking at previously seen patterns. It employs a user-specified threshold and one of the patterns as the starting leader (cluster's leader). At any step, the algorithm assigns the current pattern to the most similar cluster (if the distance between pattern and the cluster's leader is less or equal than threshold) or

the pattern itself may get added as a new leader if its similarity with the current set of leaders does not qualify it to get added to any of the existing clusters. The set of leaders found acts as the prototype set representing the clusters and is used for further decision making. Rough Set Incremental Clustering algorithm is a soft clustering method employing rough set theory [72]. It groups the given data set into a set of overlapping clusters. Each cluster is represented by a lower approximation and an upper approximation for every cluster. The lower approximation contains all the patterns that definitely belong to the cluster and the upper approximation gives overlapping area. After completion of the training of SOM on whole data set rough set incremental clustering algorithm is applied on the output nodes of SOM. After completing rough set clustering on SOM output nodes overlapped neurons and overlapped data is detected.

In the year **2009 Feyereisl et al. [38]** give an overview how SOM could be used for the different purpose of computer security. Here they described the application of SOM in the field of intrusion and anomaly detection- anomaly based, signature based, network based, host based systems. Here they have also researched SOM's capabilities of network attack and vulnerability classification, hardware security purposes like biometrics, wireless security, smart cards security, forensics purpose, clustering and categorization tool for cryptosystem attack, home gateway for intrusion detection for real time home security [73].

In the year **2010 Graham et al. [40]** told how we can visualize and compare structures of clusters using relative density of SOM nodes. Here they have shown how changes of density of SOM nodes can affect on the clustering of data set and based on change in density of nodes can find the clusters those are inherent in the data set. Like when a region in a SOM becomes significantly more dense compare to earlier SOM, and well separated from other regions, then the new region represents a new cluster.

In the year **2010 Sungwon Kim and Ki-Bum Park [69]** have developed and applied Kohonen self-organizing feature maps neural networks model (KSOFM-NNM) to forecast the daily PE for the dry climate region in south western Iran. KSOFM-NNM for Ahwaz station was used to forecast daily PE on the basis of temperature-based, radiation based, and sunshine duration-based input combinations. The measurements were used for training, cross-validation and testing data of KSOFM-NNM.

In the year **2011 Sarlin et al. [41]** applied a fuzzy clustering measure over output of SOM to find some applications of financial data, where they have considered each time-series of financial data may have fuzzy properties occupy more than one cluster, **Mount et al. [42]** quantify the boundary effects of SOM.

Khan et al. [43] applied a hybrid approach of SOM and Genetic Algorithm (GA) in stock market prediction. In past decades many researchers have applied conventional technique like Artificial Neural Network, back-propagation neural network, market basket analysis, logistic regression for forecasting and prediction of stock performance. But here they have used SOM and GA in stock market analysis which overcomes the proper weighting of criteria to obtain a list of stocks that are suitable for investments. Basically picking of stocks i.e. selection of stock are handled by SOM.

In the year **2012 Zin et al. [44]** described the 3D visualization of SOM, **Cabanes et al. [45]** described the topological learning to detect cluster, to discover imbalances in financial networks **P. Sarlin, [46] [47]** used an SOM approach, he also used SOM approach for visual tracking of development goals **[48]**.

In the year **2013 Sasanka et al. [70]** described a destructive method for the detection of crack on the basis of consideration of natural frequency. Crack in this study is transverse surface crack. In the analysis, methodologies have been developed for damage detection of a cracked cantilever beam using Kohonen network. Theoretical analysis has been carried out to calculate the natural frequency with the consideration of mass and stiffness matrices. The data obtained from theoretical analysis has been fed to Kohonen competitive learning network.

In the year **2014 Niharika et al. [71]** studied that air pollution is becoming an environmental threat with the increase in industrialization and urbanization. The air quality is becoming essential both for the environment as well to the society. There are different type of numerical as well as statistical tools for the prediction and analysis of air quality, but Artificial Neural Network is considered to be an excellent predictive and data analysis tool for Air quality forecasting.

In the year **2014 Chaudhuri et al. [49]** proposed a clustering using only SOM nearest and furthest nodes. They modified the operational structure of conventional SOM. In case of working operation of conventional SOM after competition when synaptic adaptation phase comes then all the weights of the winner and its neighbouring neurons are updated. But here after competition only farthest and nearest neurons from among the 1-neighborhood of the winner neuron being updated.

4.2 MST Based Clustering

Among several graph-theoretic based algorithms MST based clustering is very effective for high dimensional arbitrary shaped data set, where number of clusters is unknown from previous. After constructing MST on data set identify the longest edges and removing those edges based on proper cluster validity index clusters are formed.

In the year **1971 Zahn et al. [74]** introduced MST based clustering, where they describes a method to remove inconsistent edges, whose weights are significantly larger than the average weight of nearby edges in MST. Then **Asano et al. [75]** modified and described a Maximum Spanning Tree based minimization of diameter of bipartition of the tree.

In **1997 Eldershaw et al. [76]** observed the shortcomings of several clustering algorithm that it cannot find the grouping if clusters doesn't exhibit characteristics pocket and if they are very close to each other. Here data first triangulates, then partitions the edges of the resulting graph into inter- and intra-cluster edges. Using the triangulation technique neighbours

are selected for clustering. After finding the neighbours of point closest points are clustered together.

In **1997 Xu et al. [77]** applied MST in 2d-image segmentation. Here they proposed a method for partitioning gray-scale images into connected segmented regions. By constructing MST on pixel nodes and forming sub-trees from that MST and minimizing the sum of variations of gray levels over all sub-trees such that each sub-tree should have at least a specified number of nodes, and two adjacent sub-trees should have significantly different average gray-levels, segmented region of given images could be obtained. **Chowdhury et al. [78]** developed a clustering algorithm based on the MST and Bayes classifier to find the boundary between clusters.

In **2000 Lopresti et al. [79]** proposed a Euclidian-MST based RGB color clustering in image processing and analysis. This paper mainly focuses on those parts of retrieving of text from images, clustering of images and texts separately. Often we can see that an image contains some text, when we search based on those image text on search engine then how search engine can indexes the pages from www that was the main challenges of this research. For the problem of locating text in Web images, they proposed a procedure based on clustering in color space followed by a connected-components analysis. For character recognition they used polynomial surface fitting and fuzzy n-tuple classifiers.

In **2001 Xu et al. [80]** used MST for clustering of gene expression data. Here they applied MST on a set of multidimensional gene expression data. After removing inconsistent larger edges from MST sub-trees formed, where each of the sub-tree denotes each cluster of the gene expression data.

In **2005 P'aivinen [81]** used Scale Free MST (SFMST) for clustering of a scale free network.

In **2009 Grygorash et al. [82]** proposed two MST based clustering algorithms called the Hierarchical Euclidean Distance based MST clustering algorithm (HEMST) and the Maximum Standard Deviation Reduction clustering algorithm (MSDR) respectively and also **Wang et al. [83]** proposed a clustering approach using MST based on divide and conquer approach. To avoid the removing of extraneous edges without proper validation **Zhong et al. [84]** introduced two rounds of MST clustering. Basically in two different phases this clustering handles two groups of cluster problem- Separated cluster problem, touching cluster problem. In the first phase, two round MST are employed to construct a graph and detect separated clusters which cover distance separated and density separated clusters. In the second phase, touching clusters, which are subgroups produced in the first phase, can be partitioned by comparing cuts, respectively, on the two round minimum spanning trees.

In **2011 Karthikeyan et al. [85]** used MST for outlier detection and removal to produce a noise-free Meta-Similarity cluster. Here the clusters are formed hierarchically from top to bottom and optimizing number of clusters at each level. In this two phase clustering procedure first phase is responsible for creation of clusters uses divisive approach where as in the second phase dendrogram of clusters formed uses agglomerative approach.

In **2012 S. John Peter [86]** proposed a density based clustering for overlapping distribution using MST. This approach is a combined approach of density based and hierarchy based approach. In this method local density is computed at each data point using density based method, after that hierarchical approach is used by merging clusters according to the computed cluster distance based on overlap in distribution of data points.

In **2014 Chowdhury et al. [87]** used an MST based value in DBSCAN algorithm in density based clustering. To find a cluster, DBSCAN starts with an arbitrary point and retrieves all points density-reachable from that point with respect to eps-neighbourhood and MinPts. Here they proposed a modified version of DBSCAN algorithm, where they have taken the value of eps-neighbourhood by using ratio of sum of edge weights of MST and number of data point's $1/p$ coefficient; p is dimensionality of data point.

Chapter 5:

PROPOSED METHOD OF CLUSTERING

5.1 Statement of the Problem

Till now we have discussed different types of clustering algorithms in chapter 2, but most of the clustering take number of clusters as a input from previous. In that case we need a prior knowledge about number of clusters before starting of the clustering process. For example; k-means [6] is very popular clustering algorithm, faces same problem in real life. It also inefficient to separate overlapped clusters present in the data set [3]. Also in case of hierarchical agglomerative clustering we have to know where to cut the dendogram to form the cluster [7]. In the same way, the Density Based Hierarchical (DHC) clustering suffers from computational complexity in case of large data set [8]. As a result in recent era a lot of attention has been paid for creation of such an effective clustering algorithm which can handle the clustering of large or small data set that has arbitrary shape and density, without knowing actual number of cluster present in the data set from previous [9].

Keeping this view in mind we have exploited the potentiality of adaptive and competitive learning process of neural network by using self organizing property of Kohonen Self-Organizing Feature Map (SOM) for our clustering process [12]. Usually Kohonen's Self Organizing Map is applied to find the feature map. But here we have applied SOM to extracting clusters present in a given set of data points. Given a data set consisting of n number of data points, $S = \{X_1, X_2, \dots, X_n\} \in \mathcal{R}^m$ and SOM nodes $[n \times n]$, $n \in 2, \dots, 15$. Randomly initializing the nodes of SOM when data pattern is given as input on SOM, competition takes place. Based on the shortest distance between input and connection weight, competition gives winner neurons, those are represented as output nodes acted as representative data points of whole data set. By using the output nodes of SOM and with the help of application of MST on SOM output nodes we get the sub-trees of those output nodes, those helps to represents of clusters those are inherent in the whole data set.

5.2 Detailed Description of the Proposed Method

The basic idea of our proposed approach is as follows. First, we have trained a $[n \times n]$ ($n \in 2$ to 15) Self Organizing Map on whole data set. For example if we take $n=3$ then 9 ($p = n \times n = 3 \times 3 = 9$) output nodes being arranged in a two-dimensional 3 x 3 grid as shown in FIG-23. There is no guideline for choosing the appropriate value of n for a given problem. It since that the value of n should be chosen on the basis of the size of a given data set. The output nodes of the SOM are basically act as a representative nodes of the whole data set. The nodes are initialized randomly at the time of beginning the training. After starting the training when one by one data point from the data set are fed into SOM network then nodes are adjusted by taking a competition of nearest distances between data point and nodes and updating the weights $[0, 1]$ those are connecting input data point to output nodes. After competition we get winner node and only weights associated with that node are adjusted. This procedure is repeated until all data points of data set is fed into SOM network and finally we get the position of the output nodes S of training of SOM on data set.

Now we have calculated a distance matrix $z(i, j)$ using Euclidian distance measure between output nodes S of SOM. Then we have formed a graph using distance matrix between those output nodes. Then we have constructed a Minimum Spanning Tree (MST) from this graph. After that we have detected longer inconsistent edges from the MST. For detection of longer inconsistent edges we have used a procedure discussed in below.

We have set a threshold to detect longer inconsistent edges. When the weight of an edge becomes greater than threshold it becomes inconsistent edge that could be removed from MST. For this purpose we have calculated sum of edge weights of MST and dividing it with no. of edges we got the average weight W^* .

$$W^* = \frac{l_n}{n-1}$$

Where l_n = Sum of the edge weights (edge weight is taken to be the Euclidean distance) of minimal spanning tree of S .

n = Total number of data points

$n-1$ = Total number of edges of MST

We have also calculated the standard deviation σ of edge weights of MST. To get the standard deviation we have taken the square of difference of average edge weight from each edge weight of MST and taking the summation of this for all edges and then evaluated square root of this summation, which is formularized below equation----

$$\sigma = \sqrt{\frac{\sum(x_i - W^*)^2}{n-1}}$$

Where x_i is the weight of i^{th} edge of MST and W^* is the average edge weight of MST.

We have set a threshold which is the sum of average weight W^* and standard deviation of edge weight σ . There is also an alternative way of threshold selection. Threshold can also be calculated by multiplying a small value t_h with average weight W^* . The value of t_h may vary from 0.02 to 0.5 depending on the given data set. The weights of edges whose values are greater than threshold are indicated as inconsistent edges. After removing those edges from MST we have got disjoint sub-trees of the SOM nodes of MST. Those sub-trees are mainly used for creation of cluster of whole data set.

Now one by one data point taking as input, we have clustered whole data set with the help of distance based clustering. We have calculated the distances of a particular data point to all the SOM nodes those forms sub-tree. Then we have evaluated shortest distance that indicates nearest node of that particular data point. Now we have followed which sub-tree belongs to that nearest node. In case of other data points if the nearest node is the part of the same sub-tree then those data points and previous data point belongs to the same cluster. In this way whole data set are clustered.

5.3 Proposed Method in the form of an Algorithm

Input: X an $(n \times p)$ data set containing p dimensional n number of data points

Output: A set of k clusters, S_k .

Method:

- 1) Train a $[t \times t]$ Self-Organizing-Map (SOM) on dataset X
- 2) Total_no_of_SOM_nodes= $nd=t \times t$
- 3) For $i=1:nd$
- 4) For $j=1:nd$
- 5) Compute distance matrix $z(i,j)$ using Euclidean measure between output nodes of SOM
- 6) End
- 7) End
- 8) Construct an Minimum Spanning Tree (MST) from distance matrix $z(i, j)$
- 9) Compute the average weight of W of all the edges from MST
- 10) Compute standard deviation σ of the edges MST
- 11) Sort the weights of all the edges of MST in descending order
- 12) $S_T = \varphi$; $n_c = 1$; $C = \varphi$;
- 13) Repeat
- 14) For each $edge_i \in$ MST
- 15) If ($W_i > W + \sigma$)
- 16) Remove $edge_i$ from MST
- 17) If SOM node $_i$ only connected to removed edges then also remove that node
- 18) $S_T = S_T + T^{\wedge}$ // T^{\wedge} is new disjoint sub tree
- 19) $n_c = n_c + 1$
- 20) $C = \cup_{T_i \in S_T}$
- 21) End
- 22) $K=k+1$
- 23) Until (all the edges whose length $W_i > W + \sigma$ are removed)
- 24) For $i=1:n$
- 25) For $j=1:nd$
- 26) Calculate distance array $dist[i, j]$ for distance of data-point $_i$ of X with each of the node $_j \in$ SOM nodes
- 27) Calculate minimum distance $\{ \text{Min}(dist[i, j]) \}$ that represent the nearest SOM node $_j$ of that data-point $_i$
- 28) Assign the data-point $_i$ to the corresponding cluster on which the nearest SOM node $_j$ belongs
- 29) End
- 30) End

Chapter 6:

IMPLEMENTATION OF PROPOSED METHOD IN IMAGE SEGMENTATION

6.1 What Is Image Segmentation?

Segmentation is a process that partitions an image into regions. Segmentation subdivides an image into its constituent regions or objects of similar attributes. Segmentation of image is a process of partitioning a digital image into N number of parts. The images are segmented on the basis of set of pixels or pixels in a region that are similar on the basis of some homogeneity criteria like color, texture intensity, which helps to locate exact place and identify objects or boundaries in an image [91]. Let R represent the entire spatial region occupied by an image [90]. We may view image segmentation as a process that partitions R into n subregions, $R_1, R_2, R_3, \dots, R_n$, such that

$$(a) \bigcup_{i=1}^n R_i = R$$

$$(b) R_i \text{ is a connected set, } i = 1, 2, \dots, n$$

$$(c) R_i \cap R_j = \phi \text{ for all } i \text{ and } j, i \neq j$$

$$(d) Q(R_i) = \text{TRUE for } i = 1, 2, \dots, n$$

$$(e) Q(R_i \cup R_j) = \text{FALSE for any adjacent region } R_i \text{ and } R_j$$

Here $Q(R_k)$ is a logical predicate defined over the points in the set R_k , and ϕ is the null set. Two regions R_i and R_j are said to be adjacent if their union forms a connected set. First condition indicates that the segmentation must be complete that means every pixel must be in a region. Second condition says that points in a region must be connected may be 8-connected or 4-connected. Third condition indicates that regions must be disjoint. Fourth condition deal with the properties that must be satisfied by the pixels in a segmented region- $Q(R_i) = \text{TRUE}$ if all pixels in R_i have the same intensity level. Fifth condition defines that two adjacent regions R_i and R_j must be different for Q .

It is one of the most critical components of an image analysis and/or pattern recognition system and still is considered as one of the most challenging tasks in the field of image processing. It has wide applications in several domains like Medical Science, Analysis of Remotely Sensed Image, Fingerprint Recognition, and Traffic System Monitoring, Object Identification and Recognition, Criminal Investigation, Security Systems in Airport, Satellite Images, and Quality Assurance in Factories and so on.

6.2 Traditional Image Segmentation Technique

There are several approaches available for image segmentation. Traditional image segmentation methods are mainly based on the uniformity of image pixels feature values (intensity, color etc.). Thresholding is an old and simple image segmentation technique, based on the global information (e.g. Histogram of the whole image) or local information of the image. Segmentation may be obtained using edge detection of various regions. Many graph theoretic techniques also have been proposed for the purpose of image segmentation in recent years. In here image segmentation was treated as a graph partitioning problem and used normalized cut criterion which measure the inter-similarity and intra-similarity between the groups. Fuzzy C-

means also plays an important role in image segmentation recently specially in MRI image segmentation. Some traditional methods of image segmentation procedure have been discussed below.

6.2.1 **Color Based Segmentation**

An image can be segmented based on color either in RGB color space or in HSI color space. In RGB color space R denotes the Red component of image pixels, similarly G denotes Green components and B denotes Blue components of image pixels. Image can be easily converted from RGB space to HSI color space, where H represents the Hue of the image, S represents Saturation and I represent Intensity of the image. Hue is that particular color which dominates other. When we call an object is red then we basically referring its Hue i.e. red dominates other colors. Saturation refers to the relative parity or amount of white light mixed with Hue. The pure spectrum is completely saturated but pink (red + white), lavender (violet + white) are less saturated as red or violet is present with white. Intensity represents the brightness of color.

If we want to segment a particular colored region from an image in HSI color space then at first a binary mask generator generate a binary mask by thresholding the saturation image with a threshold equal to a certain amount (e.g. 5%) of the maximum value in that image. Any pixel value greater than threshold is set to 1 represents white otherwise 0 represents black. Then the product of mask image with the Hue image is taken and from the result of the product image a thresholding is done with a particular threshold value like 0.8 gives result the segmented image.

If we want to segment a particular colored region from an image in RGB color space then we have to identify which color range of image we want to segment. We have to take sample color representative points those represents average of this range and have to calculate RGB vector b that denotes this color. Now we have to classify each RGB pixel in a given image having a color in the specified range or not. To classify the pixels we have to take a similarity measure. Euclidian distance measure is the best similarity measure. By calculating the difference of Euclidian distance between two pixels RGB values and if this value comes under a particular threshold then these two pixels becomes in same color range. In this way we can get segmentation of pixels those belongs to the particular color range.

6.2.2 **Histogram Based Segmentation**

Histogram-based methods are very efficient in terms of time complexity when compared to other image segmentation methods. In this technique, a histogram is computed from all of the pixels in the image, and the peaks and valleys in the histogram are detected. Now the pixels in between two consecutive valleys can be considered to belong to a single cluster. One of the disadvantages of this method is that it is unable to work well when the image has no apparent gray level histogram peak. The other disadvantage is that the continuity of the segmented regions cannot be ensured. For the histogram based method to be efficient, we should focus on global peaks which are likely to correspond to the dominant regions in the image.

6.2.3 **Region Based Segmentation**

The region based methods are based on the similarity of pixels within a region. The first region growing method was the seeded region growing method. This method takes a set of seeds as input along with the image. The seeds mark each of the objects to be segmented. The regions are iteratively grown by comparing all unallocated neighboring pixels to the regions. The difference between a pixel's intensity value and the region's mean, δ , can be used as a measure of similarity. The pixel with the smallest difference measured this way is allocated to the respective region. This process continues until all pixels are allocated to a region. Seeded region growing requires seeds as additional input. The segmentation results are dependent on the choice of seeds. Noise in the image can cause the seeds to be poorly placed. Region splitting & merging is a modified algorithm that doesn't require explicit seeds. It starts off with a single region first, which is the image as a whole. Then the region is split into four different sub region based on some dissimilarity measures to construct a quad tree structure of regions. The regions are split to the extreme level so that no more regions splitting can be done now. Then the split regions are merged iteratively to have the final segmentation.

6.3 Segmentation of an RGB Color Image using Proposed Method

We have applied our proposed method of clustering to segment the color image to its constituent regions. The details procedure of color image segmentation using our proposed method has been discussed below.

6.3.1 Brief Discussion of Proposed Segmentation Procedure

First the image is converted into RGB dimension vector by reading the RGB values of image pixels. Then these tuples of these RGB vector are directly fed to the input layer of SOM one by one. Now an $[n \times n]$ SOM is trained on these data set consisting of RGB tuple. Then MST is constructed on the output nodes of SOM. Using pixels as the vertices of the MST, it has been used for obtaining the cluster of pixels based on their RGB values. Euclidean distance between the data points has been taken to be the edge weight of the said MST.

$$\text{Dist}(i, j) = \sqrt{(R_i - R_j)^2 + (G_i - G_j)^2 + (B_i - B_j)^2}$$

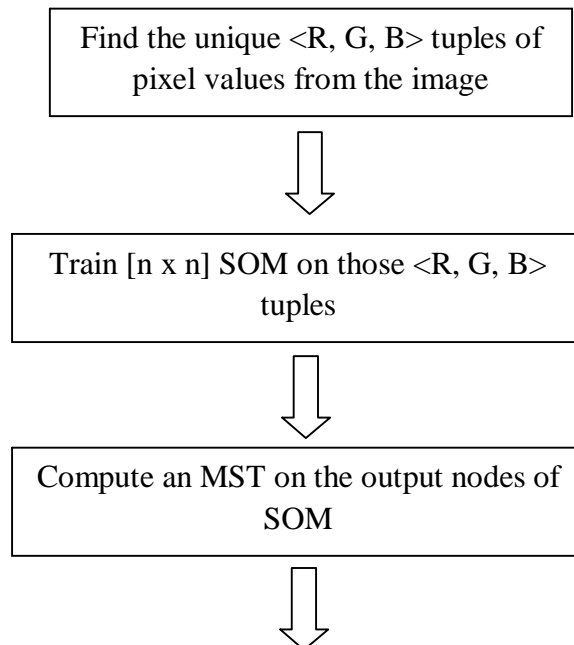
Then calculating the sum and standard deviation of edge weight of MST, we have taken a threshold as a combination of sum and standard deviation. The edges whose weights are greater than this threshold are removed from MST and form the sub-tree. Using those sub-trees we have done clustering of all RGB tuples by taking minimum distance between RGB tuples and nodes of the sub tree same as described in the proposed method of clustering. After completion of clustering of all RGB tuples with each of the cluster having different RGB values, RGB dimension vector is converted to image. This image is the segmented image of the given color image.

Now after segmentation if any noise is present due to shading of light or any other reason we have removed that noise with the help of region merging process. For removal of noise from the segmented image we have proposed an algorithm discussed in below.

```

Algo_noise_removal ( )
{
    Find the number of pixels in each region.
    If (no. of pixel in a region < 5) then
        Find the RGB values of surrounding region
        If (Total surrounding region have same RGB value) then
            Merge the region with surrounding region
            Change the RGB value of this region same as surrounding region
        Else
            Find the smallest distance between RGB tuples of this region
            and surrounding regions.
            Change the RGB value of this region same as surrounding
            region which has smallest distance.
        End If
    End If
}
    
```

6.3.2 Flow Chart of Proposed Segmentation Procedure



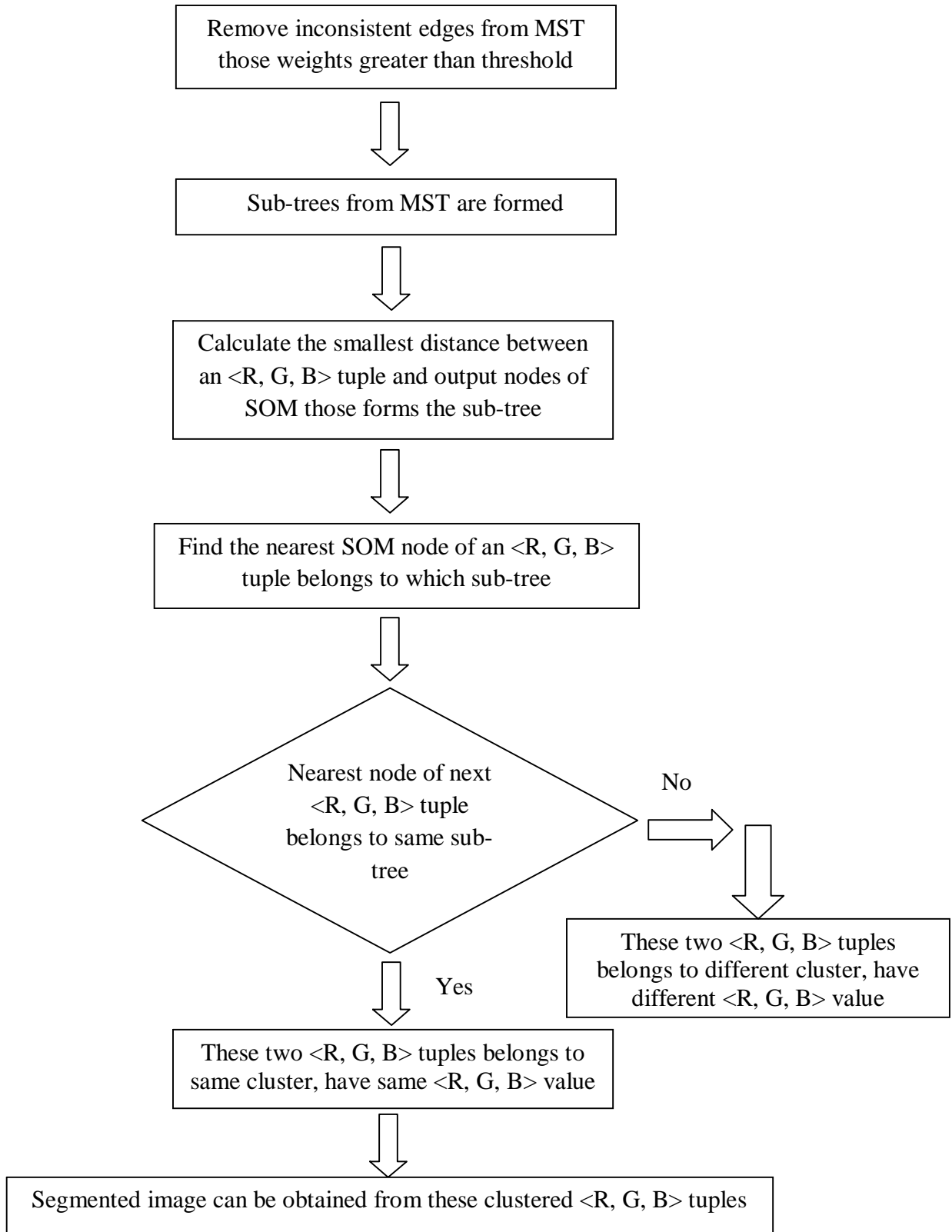


FIG-31: Outline of the Proposed Method of Image Segmentation

Chapter 7:

EXPERIMENTAL RESULTS

7.1 Software Requirements

We have done an experiment of our proposed algorithm on various Synthetic data sets and real life data sets. We have tested our algorithm on MATLAB 2013 Windows 8 platform 2GB RAM and 3.10 GHz Processor. And the necessary softwares are-----

- 1 Matlab 6.0 (Any upper version)
- 2 Windows Xp/7/8

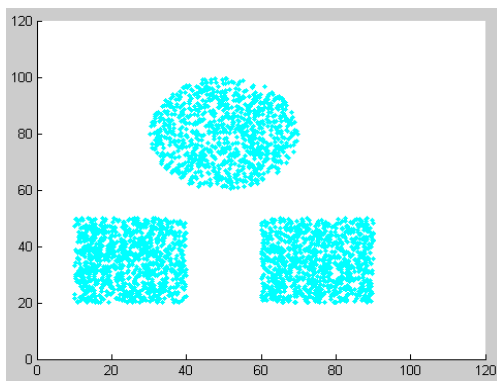
The minimum hardware requirements are:

- 1 Pentium IV/ Celeron/AMD Processor
- 2 128 MB RAM
- 3 Hard Disk (40 GB)

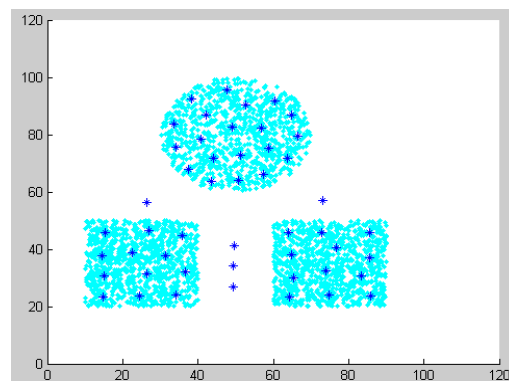
7.2 Description of the Experimental Results

7.2.1 Results on Synthetic Data Sets

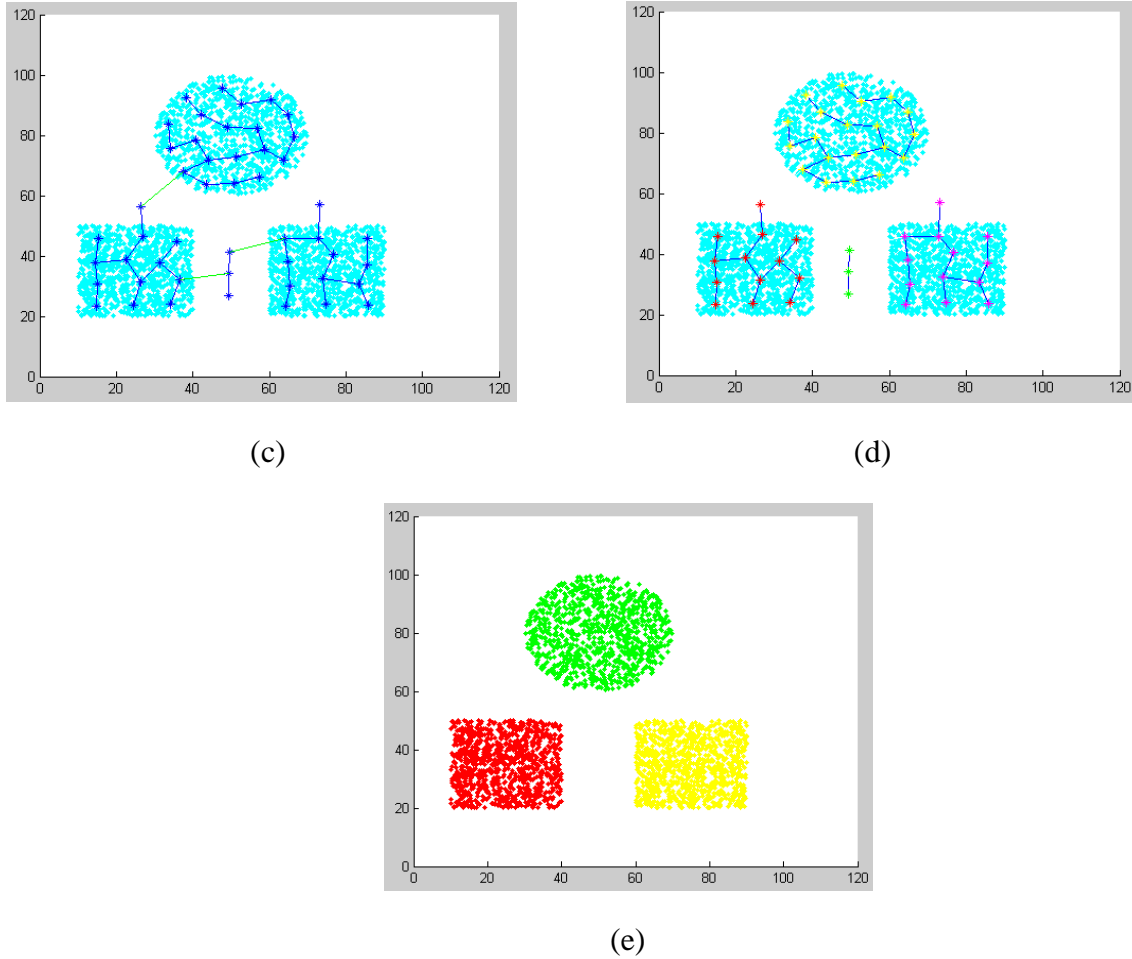
We have tested our proposed algorithms on four synthetic data sets which are shown in below figures.



(a)



(b)

**Fig. 1**

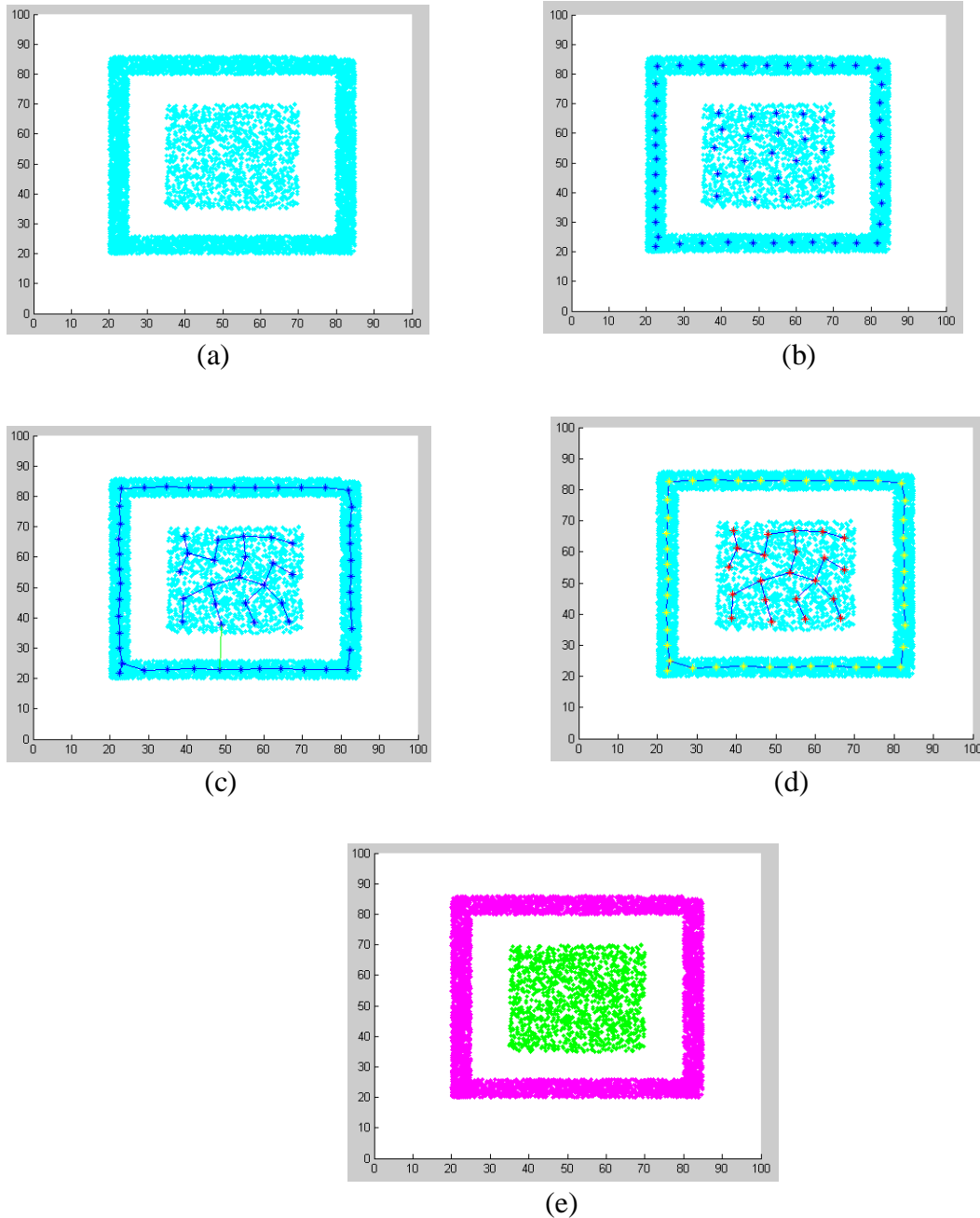
In FIG-32 a very simple data set has been shown which contains 3000 data points, those are equally divided on two squared area and a circular area, so basically each area consists of 1000 data points. The representation of this data has shown in Fig. 1 (a).

When proposed method is applied on this data set then as a result of training with [7X7] SOM, the output nodes of trained SOM are shown as blue asterisk in Fig. 1 (b).

Now after applying the Minimum Spanning Tree (MST) on the output nodes of SOM, a MST is formed with those nodes shown in Fig. 1 (c), where larger inconsistent edges detected using threshold are shown as green colour edges in the figure.

After removing those inconsistent longer edges from figure we get sub-trees of the MST, different sub-trees are marked as different colours shown in Fig. 1 (d).

Now using distance based clustering, by measuring shortest distance of data points with nodes of sub-trees all data points are clustered. Different clusters have shown in different colours, here green, red and yellow respectively shown in Fig. 1 (e).

**Fig. 2**

In FIG-28 another simple data set has been shown which contains 5000 data points, where a squared path consist of 4000 data points and within squared path a square area contains 1000 data points. The representation of this data has shown in Fig. 2 (a). After training [8X8] SOM the resulted output nodes has shown in Fig. 2 (b).The corresponding MST of nodes is in Fig. 2 (c). After removing larger inconsistent edges the resulted sub-trees having colour red and yellow have shown in Fig. 2 (d). Based on the distances of data points with nearest node of sub-tree final clustering of data points are obtained shown in Fig. 2 (e).

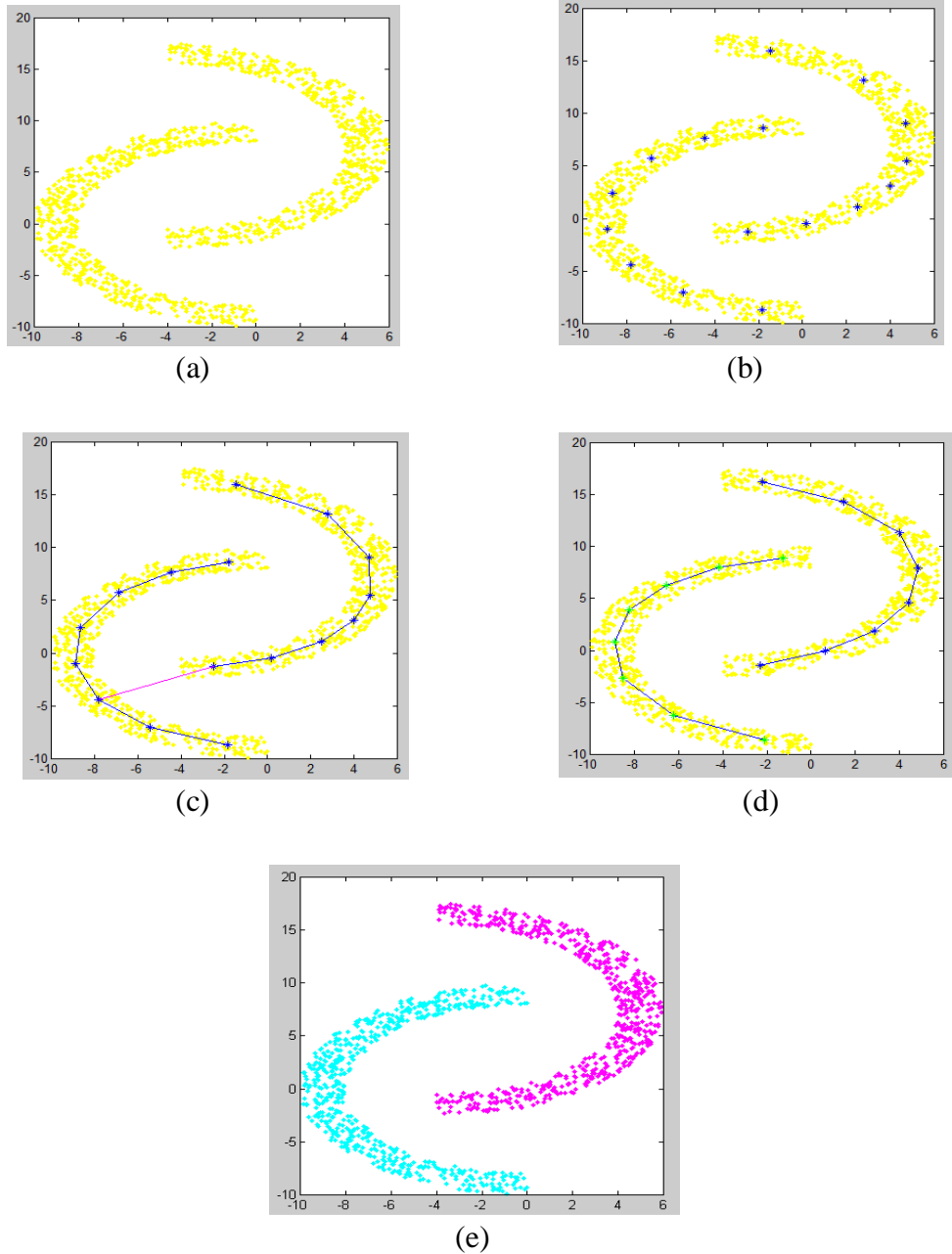
**Fig. 3**

Fig. 3 (a) shows a moon data set which consists of the two C like structure arranged in interleaved fashion, each of the structure consists of 600 data points. After applying [4X4] SOM on the whole dataset the resulted output nodes shown as in blue asterisk shown in Fig. 3 (b). The MST of the output nodes and longer inconsistent edges are shown as magenta colour in Fig. 3 (c). After removing those inconsistent edges from MST sub-trees are formed as shown in Fig. 3 (d). The result of final clustering has shown in Fig. 3 (e), where the two clusters are formed coloured magenta and cyan respectively.

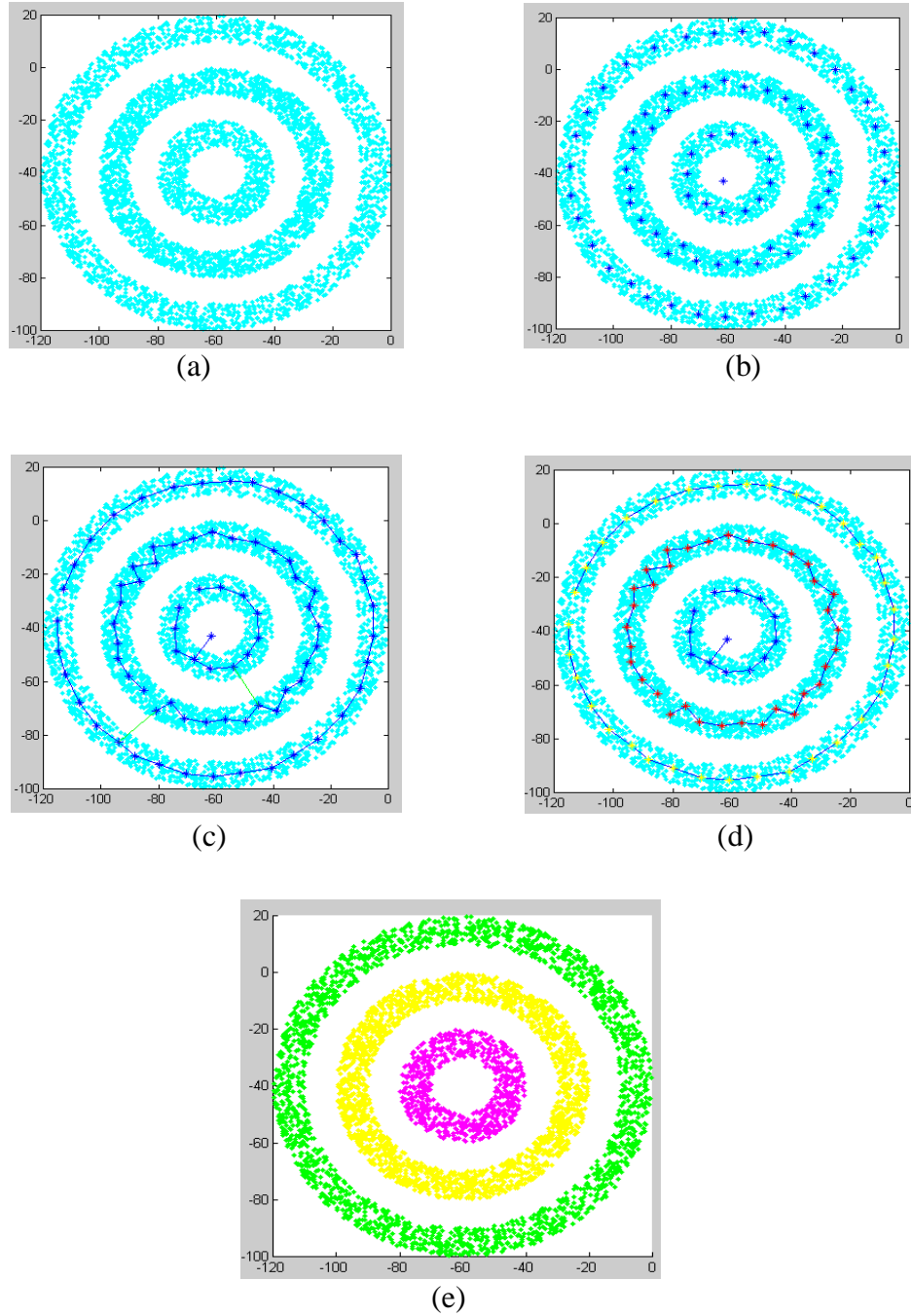
**Fig. 4**

Fig. 4 (a) shows a data set of three concentric circular rings. The data set consists of 3000 data points where each ring consists of 1000 data points. Due to the large size of data set here we have used [9X9] SOM for training. The output nodes of SOM have shown in Fig. 4 (b). After applying MST on SOM nodes and thresholding on MST edges figure becomes as in Fig. 4 (c). After removing inconsistent edges sub-trees are formed shown in Fig. 4 (d), using those sub-trees clusters of whole data set is formed shown in Fig. 4 (e).

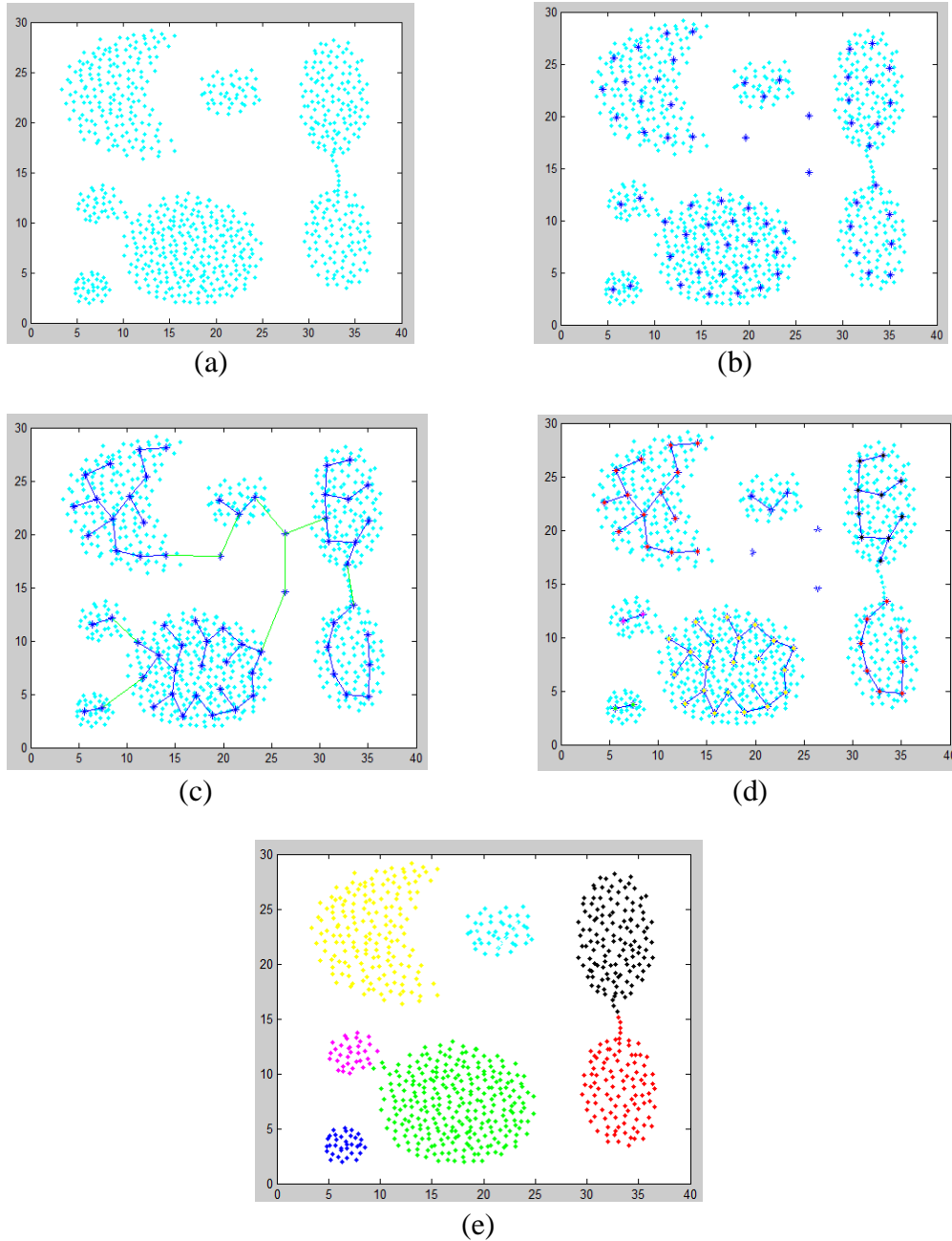
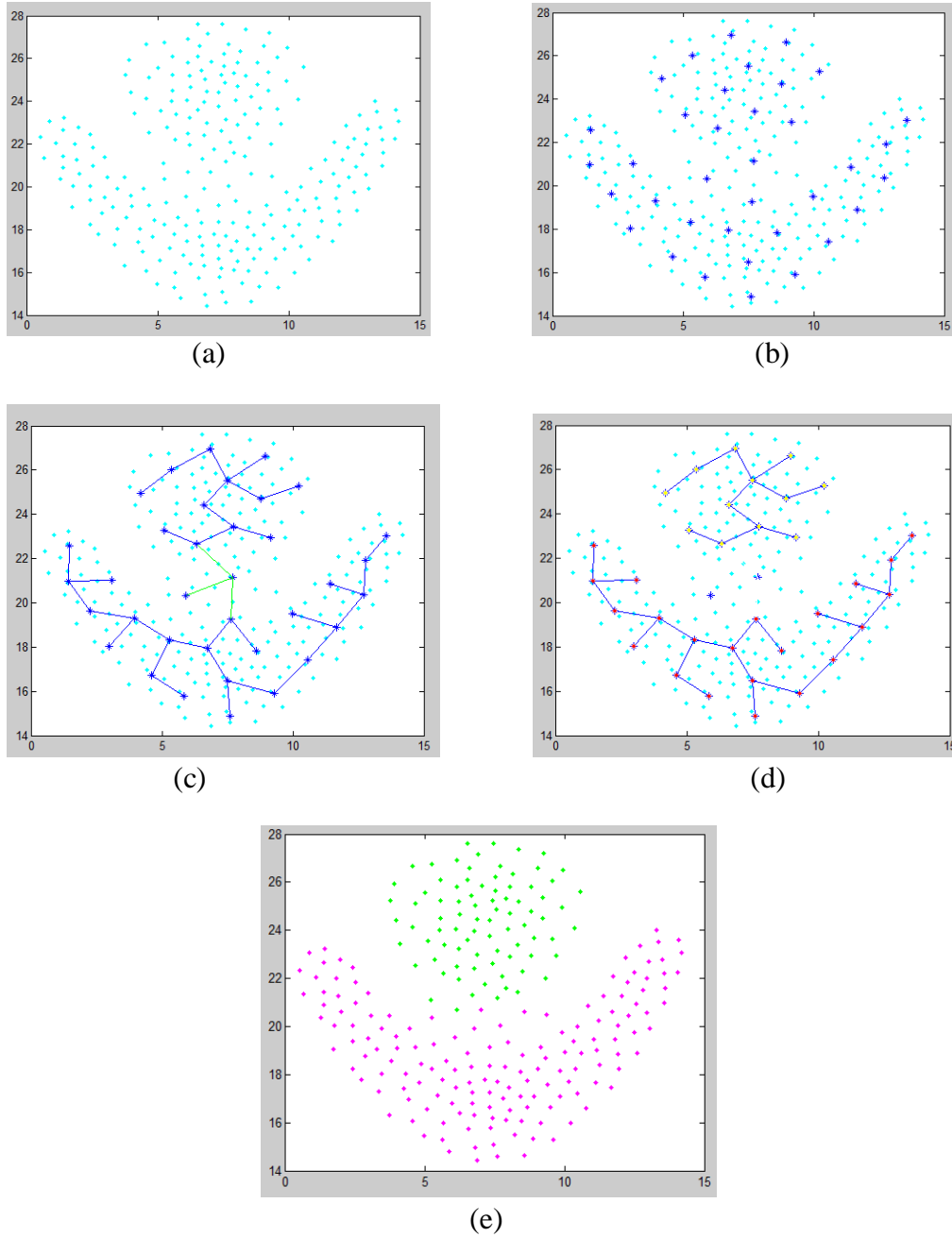
**Fig. 5**

Fig. 5 (a) is the Cluster Aggregation data set [56] that consists of seven perceptually distinct groups of points which contains 788 of data points. After training with [8X8] SOM, the positions of output nodes are presented in Fig. 5 (b). After applying MST and selecting a threshold figure becomes as Fig. 5 (c), and then removing inconsistent larger edges which are shown as green colour in (c), seven sub trees are formed which is shown in Fig. 5 (d). Seven different sub trees have seven different colors- red, yellow, green, megneta, blue, black and red respectively. Based on the distances of SOM nodes of tree with data points in the data set, whole data sets are clustered finally shown in Fig. 5 (e).

**Fig. 6**

In Fig. 6 (a) Flame data set [57] contains 240 data points where data points are distributed over two clusters. Here the whole data set is trained with [6X6] SOM. The snapshots of intermediate stages of proposed clustering -SOM output nodes representation, MST formation on output nodes, sub-tree formation from MST and final clustering of whole data set are shown as (b), (c), (d), (e) of Fig. 6 respectively.

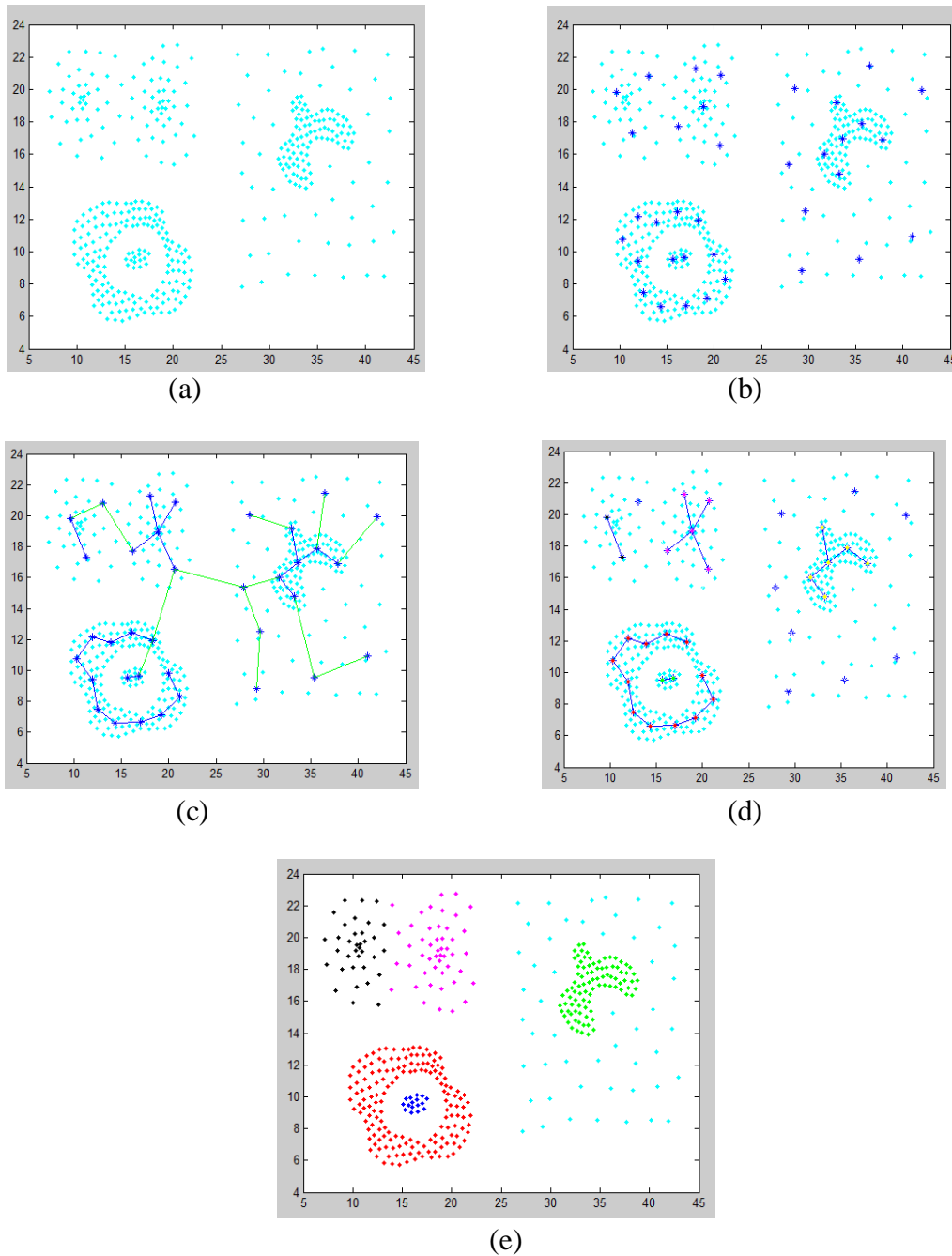
**Fig. 7**

Fig. 7 (a) shows a data set [58] that was used to detect and describe gestalt clusters that consists of 399 data points. Here the whole data set is trained with [6X6] SOM. The snapshots of intermediate stages of proposed clustering -SOM output nodes representation, MST formation on output nodes, sub-tree formation from MST and final clustering of whole data set are shown as (b), (c), (d), (e) of Fig. 7 respectively.

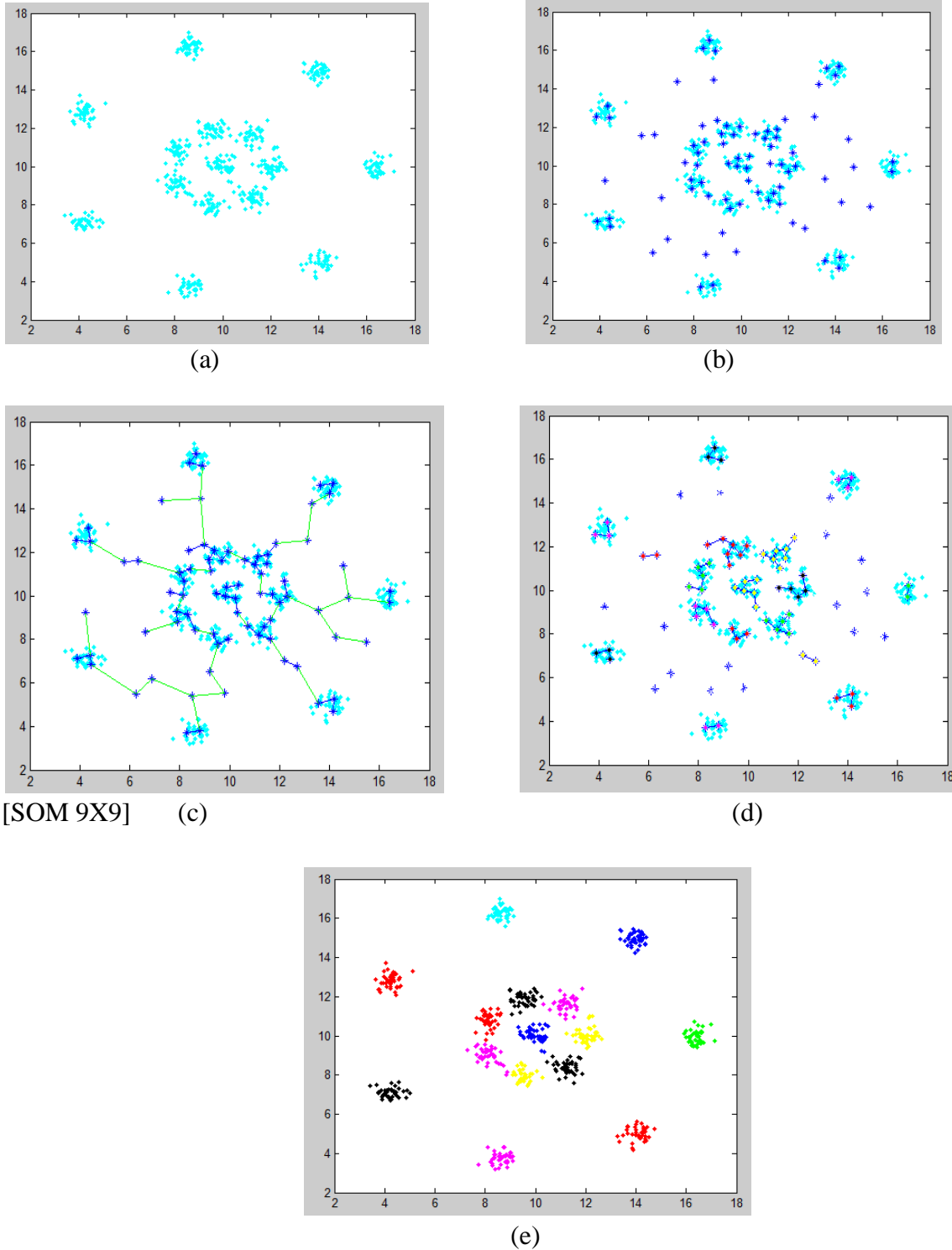


Fig. 8

Fig. 8 (a) shows a data set [59] that was used to propose a algorithm for partitional clustering that minimizes the within cluster scatter with a constraint on the cluster variance. Here

the whole data set is trained with [9X9] SOM. The snapshots of intermediate stages of proposed clustering are shown as (b), (c), (d), (e) of Fig. 8.

7.2.2 Results on Real Life Data Sets

We have done an experiment of our algorithm with two real life data sets. The proposed algorithm has been implemented on Crude-oil data [54], having 56 data points, 5 features those are basically amount of different compounds (vanadium, iron, beryllium, saturated hydrocarbon, and aromatic hydrocarbon) present in crude-oil and 3 classes has also been chosen for experimentation.

Seed data [53], a 7 dimensional data set, having 210 instances, where based on these 7 features: area, perimeter, compactness, length of kernel, width of kernel, asymmetry coefficient and length of kernel groove getting from experiment using soft x-ray technique, 3 different varieties of wheat: Kama, Rosa and Canadian, 70 elements each could be clustered.

7.3 Comparison of Results with other Clustering Methods

In comparison with DBSCAN our proposed method works well and experimental results of clustering accuracy of both shown in Table-1.

Table-1 : Comparison of Accuracy

Clustering Accuracy		
<i>Data</i>	<i>DBSCAN</i>	<i>Proposed Method</i>
Crude Oil	89.1%	92.3%
Seed	84.6%	86.9%

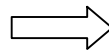
So from above table we can observe that our proposed method clustering well than traditional DBSCAN method because misclassification error in our proposed method on Crude Oil and Seed data set is 7.7% and 13.1% respectively, whereas by using DBSCAN method error became 10.9% and 15.4% respectively.

7.3 Results on Image Data Sets for Segmentation

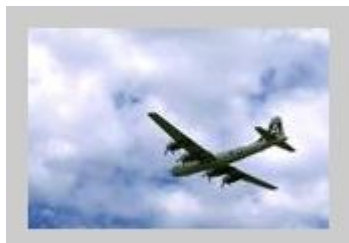
We have tested of our proposed algorithm on several images. Images are collected from the various image segmentation data base. FIG-35 shows some of the samples from the collected image database for showing experimental results. The images are color images of different resolutions. We resized the images into 100 x 150 after starting the segmentation process. The images are basically scenery images.



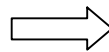
(a)



(b)



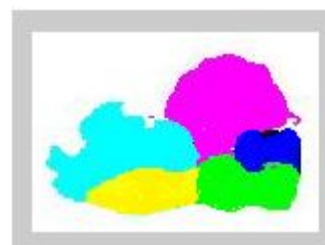
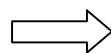
(c)



(d)



(e)



(f)

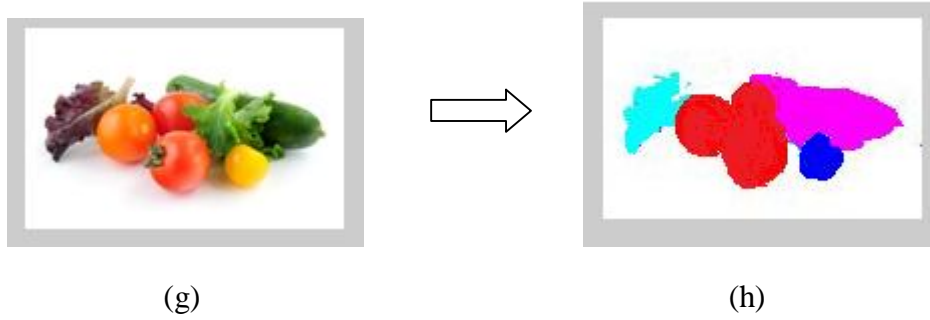


Fig. 9

Analysis in this study uses images from the Berkeley Segmentation Dataset (BSDS) [92]. There are the 4 test images whose snapshots are shown in (a), (c), (e), (g) of Fig. 9. The snapshots of corresponding segmented images are shown in (b), (d), (f), (h) of Fig. 9.

Chapter 8:
CONCLUSION
&
SCOPE FOR FURTHER RESEARCH

8.1 Conclusion

In this paper we have proposed a novel clustering algorithm based on Self-Organizing Map and minimum spanning tree, combining the gap statistic and intra-cluster average as a cluster validity index. The main objective of this work is to present such a clustering algorithm that can detect the arbitrary shaped cluster of large data set and maximum clustering accuracy. From the experiment on synthetic data sets and real life data sets it is clear that our proposed algorithm can provide cluster solution if such a natural group is present in the data set.

8.2 Scope for Further Research

There is no guideline about the numbers of nodes of the Kohonen's Self Organizing Network to be taken for a specific problem. It seems that the no. of nodes in a Kohonen's Self Organizing Network should increase with the size of input dataset. This aspect of Kohonen's Self Organizing Network may be one of the topics for further research work.

There is no proper guideline how to choose the threshold for removing edges from MST. Here we have taken sum of average and standard deviation of edge weight of MST as a threshold. It may be a topic of further research to determine the appropriate value of threshold for a given data set.

References

- [1] Palmondon R, Srihari S N. On-line and off-line handwriting recognition: A comprehensive survey[J].IEEE Transactions on Pattern Analysis and Machine Intelligence,2000,22(1):63-84.
- [2] Han, M.Kamber, Data mining: Concepts and Techniques, Morgan-Kaufman, 2006.
- [3] A. Ben-Hur and A. Elisseeff, and I. Guyon, “A Stability Based Method for Discovering Structure in Clustered Data,” Proc. Pacific Symp. Biocomputing, pp. 6-17, 2002.
- [4] A.K. Jain, M.N. Murty AND P.J. Flynn “Data Clustering: A Review”, © 2000 ACM
- [5] H. Frigui and R. Krishnapuram, “A robust competitive clustering algorithm with applications in computer vision,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 21, no. 5, pp. 450–465, May 1999.
- [6] Y. Leung, J. Zhang, and Z. Xu, “Clustering by scale-space filtering,”IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pp. 1396–1410,Dec. 2000.
- [7] A.K. Jain and R.C. Dubes, Algorithms for Clustering, prentice Hall,1988.
- [8] Meichen Yu,Arjan Hillebrand,Prejaas Tewarie,Jil Meier,Bob van Dijk,Piet Van “Hierarchical clustering in minimum spanning trees”, 2015
- [9] Zhiqiang Xie, Liang Yu, Jing Yang,A clustering algorithm based on improved minimum spanning tree, in: Proc.of Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007), 2007, pp. 149 –152.
- [10] R. Xu, D Wunsh II, Survey of clustering algorithms, IEEE Trans. on Neural Networks, 15(2005) 645-678.
- [11] C. T. Zahn, “Graph-theoretical methods for detecting and describing gestalt clusters,” IEEE Trans. Comput., vol. C-20, no. 1, pp. 68–86,Jan. 1971.
- [12] T. Mitchell, Machine Learning. New York: McGraw-Hill, 1997.
- [13] T. Kohonen.: Self-organized formation of topologically correct feature maps. Biol.Cybern. 43 59–69 (1982)
- [14] Kohonen, T.: Self-Organizing Maps. Springer, Heidelberg (2001)

- [15] R.M. Gray.: Vector quantization. IEEE Acoust. Speech, Signal Process. Mag. 1 (2) 4–29 (1984)
- [16] J. Lin, D. ye, C. Chen and M. Gao, Minimum spanning tree based special outlier mining and its applications, Lecture Notes in Computer Science, Springer-Verlag, 509 (2008) 508-515.
- [17] V. M. K. Prasad Goura, N. Madhusudana Rao, M. Rajasekhar Reddy, “A Dynamic Clustering Using Minimum Spanning Tree”, 2011 2nd International Conference on Biotechnology and Food Science IPCBEE vol.7 (2011) © (2011) IACSIT Press, Singapore.
- [18] Xiaochun Wang, Xiali Wang and D. Mitchell Wilkes, A Divide – and - conquer approach for minimum spanning Tree – based clustering, IEEE Trans. On Knowledge and Data Engg., 21(2009) 945-958.
- [19] Y. Xu, V. Olman and D. Xu, Clustering gene expression data using a graph-theoretic approach: An application of minimum spanning trees, Bioinformatics, 18(2002) 536-545.
- [20] C. Zahn. “Graph-theoretical methods for detecting and describing gestalt clusters”. IEEE Transactions on Computers, C-20:68-86, 1971.
- [21] Ramakrishnam Raju, Valli Kumari, “Comparison of Parameter Free MST Clustering Algorithm with Hierarchical Agglomerative Clustering Algorithms”, International Journal of Computer Applications (0975 – 8887) Volume 34– No.4, November 2011
- [22] MARIA HALKIDI, YANNIS BATISTAKIS , MICHALIS VAZIRGIANNIS, “On Clustering Validation Techniques”, Journal of Intelligent Information Systems, 107–145, 2001_c 2001 Kluwer Academic Publishers. Manufactured in The Netherlands.
- [23] Clustering “Non-parametric Genetic Clustering : Comparison of Validity Indices”, IEEE Trans. Systems, Man and Cybernetics Part-C, vol. 31, no. 1, pp. 120-125, 2001.
- [24] Ariel E. Baya´ and Pablo M. Granitto, “How Many Clusters: A Validation Index for Arbitrary-Shaped Clusters”, IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS, VOL. 10, NO. 2, MARCH/APRIL 2013
- [25] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the Number of Clusters in a Data Set via the Gap Statistic,” J. Royal Statistical Soc. B, vol. 63, pp. 411-423, 2003.

- [26] S. Jockusch, "A neural network which adapts its structure to a given set of patterns," in *Parallel Processing in Neural Systems and Computers*, R. Eckmiller, G. Hartmann, and G. Hauske, Eds. Amsterdam, The Netherlands: Elsevier, 1990, pp. 169–172.
- [27] J. C. Bezdek, "Some new indexes of cluster validity," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 301–315, 1998.
- [28] J. Lampinen and E. Oja. Clustering properties of hierarchical self organizing maps. *Journal of Mathematical Imaging and Vision* 2, 261–272, 1992.
- [29] F. Murtagh, "Interpreting the Kohonen self-organizing map using contiguity-constrained clustering," *Pattern Recognit. Lett.*, vol. 16, pp. 399–408, 1995.
- [30] Juha Vesanto and Esa Alhoniemi, Student Member, IEEE, "Clustering of the Self-Organizing Map", *IEEE TRANSACTIONS ON NEURAL NETWORKS*, VOL. 11, NO. 3, MAY 2000.
- [31] P. Mangiameli, S. K. Chen, and D. West, "A comparison of SOM neural network and hierarchical clustering methods," *Eur. J. Oper. Res.*, vol. 93, no. 2, Sept. 1996.
- [32] Melody Y. Kiang, "Extending the Kohonen Self-Organizing Map networks for clustering analysis", *Computational Statistics & Data Analysis* 38 (2001) 161–180
- [33] Merkl, D., Rauber, A., 2000. Uncovering the hierarchical structure of text archives by using an unsupervised neural network with adaptive architecture. *PADKK, LNAI 1805*, pp. 384–395.
- [34] Lapidot, H. Guterman, A. Cohen, Unsupervised speaker recognition based on competition between self-organizing maps, *IEEE Trans. Neural Networks* 13 (4) (2002) 877–887.
- [35] Sitao Wu, Tommy W.S. Chow, "Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density", *Pattern Recognition* 37 (2004) 175 – 188.
- [36] Y.M. Cheung, On rival penalization controlled competitive learning for clustering with automatic cluster number selection, *IEEE Trans. Knowledge Data Eng.* 17 (11) (2005) 1583–1588.
- [37] Joseph P. Herbert, Jing Tao Yao, A granular computing framework for self-organizing maps, *Neurocomputing* 9 (2009) 2865–2872.

- [38] Guilherme A. Barreto and Aluizio F. R. Araújo, "Identification and Control of Dynamical Systems Using the Self-Organizing Map", *IEEE TRANSACTIONS ON NEURAL NETWORKS*, VOL. 15, NO. 5, SEPTEMBER 2004.
- [39] Jan Feyereisl and Uwe Aickelin, "Self Organizing Maps in Computer Security", Ronald D. Hopkins et al, pp. 1-30 ISBN 978-1-60692-781-6 @ 2009 Nova Science Publishers, Inc.
- [40] M.N.M. Sap, Ehsan Mohebi, "Hybrid Self Organizing Map for Overlapping Clusters", *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2008.
- [41] Denny, W. Graham and P. Christen. Visualizing temporal cluster changes using Relative Density Self-Organizing Maps. *Knowledge and Information Systems* 25(2), 281–302, 2010.
- [42] P. Sarlin and T. Eklund. Fuzzy Clustering of the Self-Organizing Map: Some Applications on Financial Time Series. *Proceedings of the 8th International Workshop on Self-Organizing Maps (WSOM 2011)*, pp. 40–50, 2011.
- [43] N.J. Mount, D. Weaver, Self-organizing maps and boundary effects: quantifying the benefits of torus wrapping for mapping SOM trajectories, *Pattern Anal. Appl.* 14 (2) (2011) 139–148.
- [44] Stocks selected using SOM and Genetic Algorithm based Backpropagation Neural Network gives better returns", *IJCSI International Journal of Computer Science Issues*, Vol. 8, Issue 2, March 2011.
- [45] Z. Mohd Zin, M. Khalid, E. Mesbahi and R. Yusof, "Data Clustering and Topology Preservation Using 3D Visualization of Self Organizing Maps", *Proceedings of the World Congress on Engineering* 2012.
- [46] Gue'nae' l Cabanes, Youne` s Bennani, D. Fresneau, Enriched topological learning for cluster detection and visualization, *Neural Networks* 32 (2012) 186–195.
- [47] P. Sarlin. Chance discovery with self-organizing maps: Discovering imbalances in financial networks. In Ohsawa, Y., Abe, A. (eds.) *Advances in Chance Discovery*. Springer-Verlag, Heidelberg, Germany, pp. 49–61, 2012.
- [48] P. Sarlin, Z. Yao and T. Eklund. Probabilistic Modeling of State Transitions on the Self-Organizing Map: Some Temporal Financial Applications. *Intelligent Systems in Accounting, Finance and Management* 19(1), 189–203, 2012

- [49] P. Sarlin. Visual Tracking of the Millennium Development Goals with a Fuzzified Self-Organizing Neural Network. *International Journal of Machine Learning and Cybernetics* 3, 233–245, 2012.
- [50] Vikas Chaudhary , R.S. Bhatia , Anil K. Ahlawat ,” A novel Self-Organizing Map (SOM) learning algorithm with nearest and farthest neurons”,2014.
- [51] V. Pihur, S. Datta, and S. Datta, “Weighted Rank Aggregation of Cluster Validation Measures: A Monte Carlo Cross-Entropy Approach,” *Bioinformatics*, vol. 23, no. 13, pp. 1607-1615, 2007.
- [52] A.E. Baya and P.M. Granitto, “Clustering Gene Expression Data with a Penalized Graph-based Metric,” *BMC Bioinformatics*, vol. 12, article 2, 2011.
- [53] A.E. Baya and P.M. Granitto, “How Many Clusters: A validation Index for Arbitrary-Shaped Clusters”, *IEEE/ACM TRANSACTIONS ON COMPUTATIONAL BIOLOGY AND BIOINFORMATICS*, VOL. 10, NO. 2, MARCH/APRIL 2013.
- [54] UCI Machine Learning Repository, <http://archive.ics.uci.edu/ml/datasets.html>.
- [55] Johnson,R. A., and Wichern, D. W. 1982. *Applied Multivariate Statistical Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1982.
- [56] Zakariae en-naimani, mohamed lazaar, mohamed ettaouil,” Hybrid System of Optimal Self Organizing Maps and Hidden Markov Model for Arabic Digits Recognition”, *WSEAS TRANSACTIONS on SYSTEMS*, Volume 13, 2014.
- [57] Gionis, A., H. Mannila, and P. Tsaparas, Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2007. 1(1): p. 1-30.
- [58] Fu, L. and E. Medico, FLAME, a novel fuzzy clustering method for the analysis of DNA microarray data. *BMC bioinformatics*, 2007. 8(1): 3.
- [59] Zahn, C.T., Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on Computers*, 1971. 100(1): p. 68-86.
- [60] Veenman, C.J., M.J.T. Reinders, and E. Backer, A maximum variance cluster algorithm. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2002. 24(9): p. 1273-1280.
- [61] Chau-Yun Hsu and Hwai-En Wu “An Improved Algorithm for Kohonen's Self-organizing Feature Maps”, 0-7803-0593-0/92 1992,IEEE.

- [62] N Vassilas , P Thiran and P Ienne “HOW TO MODIFY KOHONEN’S SELF-ORGANISINGFEATURE MAPS FOR AN EFFICIENT DIGITAL PARALLEL IMPLEMENTATION” ‘Artificial Neural Networks’, 25-28 June 1995, Conference Publication No. 409,O IEEE, 7995.
- [63] Kothari ,R. and Islam, S.(1996) “Spatial Characterization of Remotely Sensed Soil Moisture Data Using Self Organizing Feature Maps” IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, VOL. 37, NO. 2 MARCH 1999.
- [64] Kikuo Fujimura et al. “The Automatic Button-color Matching System using Kohonen’s Self- Organizing Feature Maps in the Textile Field” 0-7803-3210-5/96, 1996 IEEE.
- [65] Kikuo Fujimura et al. “Application of Kohonen's Self-organizing Feature into the Problem of Selecting the Buttons” Proceedings of 1993 International Joint Conference on Neural Networks.
- [66] Allamehzadeh, M. and Mokhtari, M(2003) “Prediction of Aftershocks distribution Using Self- Organizing Feature Maps (SOFM) and I ts Application on the Birjand-Ghaen and I zmit Earthquakes” JSEE: Fall 2003, Vol. 5, No. 3 / 1
- [67] Moradkhani,H., Hsu K., Gupta,H.V., Sorooshian, S. (2004) "Improved streamflow forecasting using self-organizing radial basis function artificial neural networks"Journal of Hydrology 295 (2004) 246–262
- [68] Madan,A.(2005)"Vibration control of building structures using self-organizing and self-learning neural networks"Journal of Sound and Vibration 287 (2005) 759–784
- [69] Sen, S. et al. “Species Classification Using DNA-Sequences By Self-Organizing Feature Map (SOFM)” Proceedings of the International Conference on Resource Utilization and Intelligent Systems, Kongu Engg. College, Perundurai, Erode, T.N., India. Jan. 4 - 6, 2006. pp.1 - 8.
- [70] Kim, S. and · Park, K “Application of Soft Computing Model for Hydrologic Forecasting”
- [71] Choudhury,S and Parhi, D (2013)"Crack detection of a cantilever beam using kohonen network techniques" Global Advanced Research Journal of Engineering, Technology and Innovation (ISSN: 2315-5124) Vol. 2(5)pp. 153-157, June, 2013

- [72] Niharika, Venkatadri M, Rao, P.S. (2014) "A survey on Air Quality forecasting Techniques" International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 103-107
- [73] Pawlak, Z.: Rough sets. *Internat. J. Computer Inf. Sci.* (11) 341–356 (1982)
- [74] Hayoung Oh, Jiyoung Lim, Kijoon Chae, and Jungchan Nah, Home gateway with automated real-time intrusion detection for secure home networks, *Computational Science and Its Applications - ICCSA 2006, LNCS, 2006*, pp. 440–447.
- [75] C. Zahn. "Graph-theoretical methods for detecting and describing gestalt clusters." *IEEE Transactions on Computers*, C-20:68–86, 1971.
- [76] T. Asano, B. Bhattacharya, M. Keil, and F. Yao. Clustering algorithms based on minimum and maximum spanning trees. In *Proceedings of the 4th Annual Symposium on Computational Geometry*, pages 252–257, 1988.
- [77] C. Eldershaw and M. Hegland. Cluster analysis using triangulation. In B. Noye, M. Teubner, and A. Gill, editors, *Computational Techniques and Applications: CTAC97*, pages 201–208. World Scientific, 1997.
- [78] Y. Xu and E. Uberbacher. 2d image segmentation using minimum spanning trees. *Image and Vision Computing*, 15(1):47–57, 1997.
- [79] N. Chowdhury and C.A. Murthy, "Minimum Spanning Tree Based Clustering Technique: Relationship with Bayes Classifier", *Pattern Recognition*, 30(11), 1997, 1919-1929.
- [80] D. Lopresti and J. Zhou. Locating and recognizing text in www images. *Information Retrieval*, 2:177–206, 2000.
- [81] Y. Xu, V. Olman, and D. Xu. Minimum spanning trees for gene expression data clustering. *Genome Informatics*, 2:24–33, 2001.
- [82] N. P. Aivinen. Clustering with a minimum spanning tree of scale-free-like structure. *Pattern Recogn. Lett.*, 6(7):921–930, 2005.
- [83] O. Gryorash, Y. Zhou and Z. Jorgenssn, "Minimum Spanning tree-based Clustering Algorithms", *Proc. IEEE Int'l Conf. Tools with Artificial Intelligence*, 2006, pp. 73-81.
- [84] Xiaochun Wang, Xiali Wang and D. Mitchell Wilkes, "A Divide-and-conquer Approach for Minimum Spanning Tree-based Clustering", *IEEE Transactions on Knowledge and Data Engg.*, 21, 2009.

- [85] Zhong, C. and Miao, D. and Wang, R. A graphtheoretical clustering method based on two rounds of minimum spanning trees. *Pattern Recognition*, 43, no. 3 (2010): 752-766.
- [86] T. Karthikeyan, S. John Peter,” Meta Similarity Noise-free Clusters Using Dynamic Minimum Spanning Tree with Self-Detection of Best Number of Clusters”, *Journal of Emerging Trends in Computing and Information Sciences*, Volume 2 No.4, APRIL 2011 ISSN 2079-8407.
- [87] S. John Peter,” Local Density-based Hierarchical Clustering for Overlapping Distribution using Minimum Spanning Tree”, *International Journal of Computer Applications* (0975 – 8887) Volume 43– No.12, April 2012.
- [88] Nirmalya Chowdhury, Preetha Bhattacharjee, “Using an MST based Value for ϵ in DBSCAN Algorithm for Obtaining Better Result”, *I.J. Information Technology and Computer Science*, 2014, 06, 55-60.
- [89] S. Nagendrudu, V.Ramakrishna Reddy, “Enhanced Clustering of High Dimensional Data Using Fast Cluster Based Feature Selection”, *IJCSET*, May 2015 , Vol 5, Issue 5,113-115.
- [90] S. Guha, R. Rastogi, and K. Shim, 2000. ROCK: A Robust Clustering Algorithm for Categorical Attributes. *Information Systems*, vol. 25, no. 5 : 345-366.
- [91] Rafael C. Gonzalez, Richard E Woods “Digital Image Processing” 3rd Edition.
- [92] Rajeshwar Dass, Priyanka, Swapna Devi “Image Segmentation Techniques” *IJECT*, Vol. 3, Issue 1, Jan- Mar 2012, pp 66-70.
- [93] www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench.
- [94] Anil K. Jain, Robert P.W. Duin, “Introduction to Pattern Recognition”.
- [95] R. C. Gonzalez, “Object Recognition,” in *Digital image processing*, 3rd ed. Pearson, August 2008, pp. 861-909.
- [96] Sergios Theodoridis, Konstantinos Koutroumbas, “Pattern Recognition”4th ed.
- [97] S. Kotsiantis, Supervised Machine Learning: A Review of Classification Techniques, *Informatica Journal* 31 (2007) 249-268 (http://www.informatica.si/PDF/313/11_Kotsiantis%20%20Supervised%20Machine%20Learning%20-%20A%20Review%20of...pdf).
- [98] Aldenderfer, Mark S., & Blashfield, Roger K. (1984). *Cluster Analysis*. Sage Publications, Newbury Park, Cal. on page 13.

- [99] Lotfi A. Zadeh, “Fuzzy Logic, Neural Networks, and Soft Computing”, 1994 ACM Communication Vol.37.No.3.
- [100] R.P. Lippmann, “An Introduction to Computing with Neural Nets”, IEEE ASSP Magazine, Vol. 4, No. 2, Apr. 1987, pp. 4-22.
- [101] J. Mendel. Fuzzy logic systems for engineering: a tutorial. Proceedings of the IEEE, 83(3):345-377, Mar 1995.
- [102] A.K. Jain, M.N. Murty AND P.J. Flynn “Data Clustering: A Review”, ©2000 ACM
- [103] J.Han, M.Kamber, Data mining: Concepts and Techniques, Morgan-Kaufman, 2006.
- [104] G. Cybenko, “Approximation by superpositions of a sigmoidal function,” Math. Contr. Signals Syst., vol. 2, pp. 303–314, 1989.
- [105] K. Hornik, “Approximation capabilities of multilayer feedforward networks,” Neural Networks, vol. 4, pp. 251–257, 1991.
- [106] Baez, P. G., Araujo, C. S., Fernandez, V. & Procházka, A. [2011]. Differential Diagnosis of Dementia Using HUMANN-S Based Ensembles, Springer, Berlin, Germany, chapter 14, pp. 305–324.
- [107] Kohonen, T. [1989]. Self-Organization and Associative Memory, Springer-Verlag.