CERTIFICATE

This is to certify that my work presented in this thesis entitled **"Design and development of an algorithm for fuzzy entropy"** submitted by **Mr. Amit Agrawal** in partial fulfillment of the requirement for the award of the degree of **Master of Engineering in Electronics Instrumentation and Control Engineering at Thapar university (Deemed University), Patiala**, is an original record under supervision and guidance of **Dr Yaduvir Singh & Mrs. Gagandeep Kaur**. The matter embodied in this report has not been submitted anywhere for the award of any degree.

Date:

Amit Agrawal
Roll No 80651001

It is certified that the above statement made by the student is correct to the best of our knowledge and belief.

Dr. Yaduvir Singh
Assistant Professor, EIED
(Supervisor)
Thapar university, Patiala

Mrs. Gagandeep kaur
Senior Lecturer, EIED
(Co-supervisor)
Thapar university, Patiala

Dr. Samarjeet Ghosh
Professor& Head, EIED
Thapar University, Patiala

Dr. R.K. Sharma
Dean of Academic Affairs
Thapar University, Patiala

Dedicated to my parents

ACKNOWLEDGEMENT

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life. My greatest thanks are to my parents who bestowed ability and strength in me to complete this work. I am deeply indebted to my parents and friends for their inspiration and ever encouraging moral support, which enabled me to pursue my studies.

I am very thankful to the Head of the Department, Dr. Samarjeet Ghosh, for his encouragement, support and for providing the facilities for the completion of this thesis.

This work would not have been possible without the encouragement and able guidance of my supervisor Dr Yaduvir Singh. His enthusiasm and optimism made this experience both rewarding and enjoyable. Most of the novel ideas and solutions found in this thesis are the result of our numerous stimulating discussions. His feedback and editorial comments were also valuable for writing this thesis. Also I shall be failing in my duties if I do not express my deep sense of gratitude towards Mrs. Gagandeep kaur who has been a constant source of inspiration for me throughout this wok.

I am also very thankful to the entire faculty and staff members of Electrical Instrumentation Department for their direct–indirect help, cooperation, and love affection which made my stay at T.U. memorable.

Date:

Amit Agrawal
(ROLL NO. 80651001)

ABSTRACT

A measure is developed for measuring the amount of information given when the characterizing function of a fuzzy set is only partly specified. Its modification is considered when an aprior characterizing function for the set is also given. For a fuzzy set, we may not given the values of all of $\mu_A(X_1)$, $\mu_A(X_2)$……. $\mu_A(X_n)$, but we may give some partial information about these in the form of equality or inequality relation between the values of these. We have given a method for measuring the information provided by each of these pieces of knowledge. This knowledge will change if some prior information based on intuition or experience is available about the possible values of these membership functions. We have considered here how this information is modified in this case. Finally we have taken a general situation when we have measured some partial knowledge given about n positive real numbers and we have evaluated the information contained in this partial knowledge.

This thesis deals with probabilistic measures of information. A large number of measures of probabilistic information have been developed during the last five decades. Probabilistic measures of fuzzy information include fuzzy entropy, fuzzy directed divergence, fuzzy distance, fuzzy total ambiguity etc. Fuzzy uncertainty is different from probabilistic uncertainty. Fuzzy entropy measures uncertainty due to fuzziness of information, while probabilistic entropy measures uncertainty due to the information being available in terms of a probability distribution only. A close link has been established between measure of information for probabilities and fuzzy set cases. This a step in the direction of integrating these two approaches to understand uncertainty.

In this thesis incomplete quantitative data has been dealt by using the concept of fuzzy entropy. Genetic programming has been used to classify the incomplete data. Certain attributes related to the data have been considered. Test data used in this knowledge discovery algorithm knows the entire attribute clearly. The developed algorithm is very effective and can be used in the various application related to knowledge discovery and machine learning. The developed knowledge discovery algorithm using fuzzy entropy has been tested for verity of incomplete data sets pertain to various application and it is found that the error level is merely $\pm 4.40\%$, which is far better than other available knowledge discovery algorithms.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

- AI Artificial Intelligence
- FL Fuzzy Logic
- MF Membership Function
- FE Fuzzy Entropy
- FS Fuzzy Set
- DC Data Classification
- FCM Fuzzy C- means
- CP Cluster Partition
- GP Genetic Programming
- SL Supervised Learning
- FC Fuzzy Classifier

List of Appendices

LITERATURE SURVEY

Yizong Cheng [1] et.al introduces a family of fuzzy clustering algorithms that are iterations based on alternate membership evaluations and cluster center shifts are compared with the blurring process, a deterministic dynamic system that moves data points to weighted means in their neighborhoods. It is shown in this paper that when the initial cluster centers are assigned as data points themselves, some fuzzy clustering algorithms, particularly the maximum-entropy clustering, become blurring processes. Some basic results obtained in the blurring process thus can be applied to these special runs of fuzzy clustering, and may serve as counterexamples for fuzzy clustering in general.

M. Delgado [2] et.al explains Fuzzy clustering used for identification of (fuzzy) systems. Starting from a set of examples (input-output pairs) of a certain system, fuzzy clustering permits to disclose fuzzy rules driven the given system and also to make direct inference from new observations of the input. This paper attempts to present an approach to the problem of validating fuzzy clustering processes. The cluster method before the fuzzy clustering, in order to select a suitable initial structure. With this objective, several consistence measures for crisp classifications are introduced. Using these measures on the hierarchy of classifications associated to an hierarchical cluster the most suitable level is obtained. From this classification the fuzzy clustering process is started.

Mika Sato [3] et.al presents a clustering problem in which the observations of the objects are given by the values involving vagueness; the ordinary fuzzy clustering methods are not available. In this paper, these data are treated as fuzzy data which are defined by convex and normal fuzzy sets (CNF sets), and a new fuzzy clustering model for the fuzzy data is proposed. We define a conical membership function to represent the CNF sets, and propose a fuzzy dissimilarity between a pair of fuzzy observations, which is an extension of the fuzzy This dissimilarity, discussed in this paper, becomes asymmetric. Therefore, we obtain two different clustering results with respect to each asymmetric part. To achieve consistent clustering results, using an additive fuzzy clustering model, we will solve by a multi criteria clustering technique.

Jianhua Chen [4] et.al explores the application of a fuzzy clustering algorithm in the field of Chemical Process Control. Process Control is fundamental to Industrial Chemical Operations. The control problem considered in this study is a two level cascade control of the pH in a chemical stream. The pH is controlled by addition of two chemicals – Sulfuric acid (to lower the pH) and Caustic (to increase the pH). The fuzzy clustering algorithm is used in this study to identify fuzzy rules from numerical I/O data points. The algorithm replaces the notion of a single representative point of a cluster with a more general notion of a hyper plane for each cluster. In this study, a simulation of the control problem has been generated and a menu-driven GUI has been developed which enables the user to simulate different states of the control problem by modifying the tuning parameters. Preliminary experiments show that the rules learned by the fuzzy clustering method perform well. These results provide support for the use of fuzzy clustering algorithms in process control.

F. Klawonn [5] et.al describes techniques for deriving fuzzy if-then rules based on special modified fuzzy clustering algorithms. The basic idea is that each fuzzy cluster induces a rule. The fuzzy sets appearing in a rule associated with a fuzzy cluster are obtained by projecting the cluster to the one dimensional coordinate spaces. In order to allow clusters of varying shape and size we derive special fuzzy clustering algorithms which are searching for clusters in the form of axes-parallel hyper-ellipsoids.

Chen, J [6] et.al proposed a two-step method for designing fuzzy rules when no plant model or control surface table is available. The first step learns heuristic fuzzy rules by performing online adaptive control via a trial-and-error method. One simple rule is to choose the control y(t) such that both the plant-state error x(t) and the change of error <x(t) move toward zero at the same rate, up to some constant factor. The second step applies fuzzy clustering to the rule data generated by the first step to obtain more general and robust fuzzy control rules. Our experiments with an inverted pendulum problem show a good performance.

Chih-Hsiu Wei [7] at .al has proposed Fuzzy clustering (c-means) which is a widely known unsupervised clustering algorithm, but it can not guarantee to find the global minimum, because it approximates the minimum of an objective function by the iterative method in solving the

differentiation problem, starting from a given point. For overcoming this drawback, we incorporate the genetic search strategies in the fuzzy clustering algorithm to explore the data space from a multiple-point concept. The direct application of the genetic algorithms to the fuzzy clustering is not suitable, because sometimes the data set is enormous. Under this situation, the chromosome would be too long, so a distributed approach to fuzzy clustering by genetic algorithms is proposed to divide the huge search space into many small ones.

Linkens, D.A [8] et.al proposes a fast and computationally efficient fuzzy clustering approach is presented. In this approach, fuzzy clustering is implemented in two hierarchical phases: sub clusters generation by a self-organizing network and fuzzy classification via a fuzzy competitive clustering network associated with a fuzzy c-means algorithm. Owing to the hierarchical network, the computation complexity of fuzzy clustering is reduced drastically and the clustering performance is enhanced as well. The simulation results show that the proposed method has a much higher computing efficiency and better classification performance compared to standard fuzzy c-means clustering.

Miyamoto, S [9] et.al reviewed principal methods in nonhierarchical and hierarchical fuzzy clustering. In particular, the method of fuzzy c-means is focused upon and recent algorithms in fuzzy c-means are described. It is shown that the concept of regularization plays an important role in the fuzzy c-means. Classification functions induced from fuzzy clustering are discussed and variations of the standard fuzzy c-means are introduced. The hierarchical classification based on the transitive closure is equivalent to the single link method of agglomerative clustering. The roles of the concept of fuzziness in nonhierarchical and hierarchical methods are thus contrasted.

Kreinovich [10] et.al discusses many real-life decision-making situations; in particular, in processing satellite images, we have an enormous amount of information to process. To speed up the information processing, it is reasonable to first classify the situations into a few meaningful classes (clusters), find the best decision for each class, and then, for each new situation, to apply the decision which is the best for the corresponding class. One of the most efficient clustering methodologies is fuzzy clustering, which is based on the use of fuzzy logic. Usually, heuristic clustering are used, i.e., methods which are selected based on their empirical efficiency rather

than on their proven optimality. Because of the importance of the corresponding decision-making situations, it is therefore desirable to theoretically analyze these empirical choices. In this paper, we formulate the problem of choosing the optimal fuzzy clustering as a precise mathematical problem, and we show that in the simplest cases, the empirically best fuzzy clustering methods are indeed optimal.

Tai Wai Cheng [11] et.al presents a multistage random sampling fuzzy c-means based clustering algorithm, which significantly reduces the computation time required to partition a data set into c classes. A series of subsets of the full data set are used for classification in order to provide an approximation to the final cluster centers. The quality of the final partitions is equivalent to that of fuzzy c-means. The speed-up is normally a factor of 2-3 times, which is especially significant for high dimensional spaces and large data sets. Examples of the improved speed of the algorithm in two multi-spectral domains, magnetic resonance image segmentation and satellite image segmentation, are given. The results are compared with fuzzy c-means in terms of both the time required and the final resulting partition. We show speed-up results from the application of fuzzy clustering to fuzzy rule generation in the domain of magnetic resonance imaging. Significant speed-up is shown in each example presented in the paper. Further, the convergence properties of fuzzy c-means are preserved.

Altman, D [12] et.al proposes Cluster analysis involving the use of a class of algorithms that aim to partition a data set into a number of disjoint groups. These groups have the property that members within a group are more similar to each other than members from different groups. Cluster analysis is used to examine underlying patterns or groupings in data. In remote sensing it can be used to determine the natural spectral classes in a data set and hence to select representative training samples for classification. A feature of many data sets in the Earth sciences is the presence of observations that exhibit characteristics of more than one class. While this may be overcome by collecting data at a higher resolution, it is not always feasible or possible to do this Fuzzy clustering is a variant of conventional clustering that overcomes the limitations of samples being committed wholly to one group. In fuzzy clustering, a fuzzy set is used to represent a sample's multiple group membership. Each element of the set is the degree of belonging of the sample to a particular group. This is a numerical value with range [0,1] such

that the sum of all group memberships for a sample is exactly 1. This paper presents two operational improvements that can reduce the execution time of a standard fuzzy clustering algorithm, the Fuzzy c-Means (FCM) algorithm. An FCM algorithm is an iterative partitioning method that produces optimal c-partitions. A fuzzy partition is an array of N fuzzy membership sets that are created when a set of N samples in n-space are fuzzy clustered into c groups.

A.K. Jain [13] et.al has proposed Data Clustering, a review which talks about talks about various clustering techniques Clustering, which is the unsupervised classification of patterns (observations, data items, or feature vectors) into groups (clusters) has been addressed in many contexts and by researchers in many disciplines; reflecting its broad appeal and usefulness as one of the steps in exploratory data analysis. However, clustering is a difficult problem combinatorial, and differences in assumptions and contexts in different communities have made the transfer of useful generic concepts and methodologies slow to occur. This paper presents an overview of pattern clustering methods from a statistical pattern recognition perspective, with a goal of providing useful advice and references to fundamental concepts accessible to the broad community of clustering practitioners. Taxonomy of clustering techniques is identified, and crosscutting themes and recent advances are described along. Some important applications of clustering algorithms such as image segmentation, object recognition, and information retrieval are also described.

Tormod Naes [14] et.al has proposed a discussion of the versatility and flexibility of fuzzy clustering. Three examples of very different applications are presented. The focus is on flexibility with respect to distance measure used and with respect to the possibility of utilizing known membership values for some of the samples.

P.J. Costa [15] et.al introduces the noise is source of ambiguity for fuzzy systems. Although being an important aspect, the effects of noise in fuzzy modeling have been little investigated. This paper presents a set of tests using three well-known fuzzy modeling algorithms. These evaluate perturbations in the extracted rule-bases caused by noise polluting the learning data, and the corresponding deformations in each learned functional relation. Effect of noise on fuzzy modeling systems is presented and fuzzy model structure influences noise sensitivity of each

algorithm is calculated along with characteristics of the learning algorithms which are relevant to noise attenuation.

Keller, A.[16] et.al introduces an objective function-based fuzzy clustering technique that incorporates linear combinations of attributes in the distance function. The main application field of our method is image processing where a comparison pixel by pixel is usually not adequate, but the environment of a pixel or groups of pixels characterize important properties of an image or parts of it. In addition, our approach can be seen as generalization of other fuzzy clustering techniques like the axes-parallel version of the Gustafson-Kessel algorithm.

Geva, A.B [17] et.al proposes that applying clustering analysis to sliding windows of non-stationary time-series is useful for grouping related temporal patterns that are dispersed along the time-series. Since the input patterns are time-series, a similar series of events that lead to a similar result would be clustered together. The switches from one stationary state to another (changes of regime), which are usually vague and not focused on any particular time point, are naturally treated by means of fuzzy clustering. In the first stage of the method, the time-series is rearranged into sliding windows of temporal patterns. In the next stage, similar temporal patterns are grouped together into clusters, which may represent the different states of the dynamic system, by an unsupervised fuzzy clustering procedure. A time-series prediction model is fitted to each cluster separately using its similar past temporal patterns as a training set. In the last stage, the future samples of the time-series are predicted by a fuzzy mixture of the above prediction models weighted by the degree of membership of the latest temporal pattern in each of the corresponding clusters. The hybrid algorithm suggested for the clustering is a hierarchical version of the unsupervised optimal fuzzy clustering algorithm. One of the advantages of this new algorithm is its adaptive hierarchical selection of the number of clusters (the number of underlying processes, or states, in the time-series), which can overcome the general non-stationary nature of real-life time-series (biomedical, physical, economical, etc.). The method is demonstrated for well-known time-series benchmarks.

Russell, S [18] et.al explores fuzzy clustering approaches to telecommunications database marketing. Fuzzy clustering methods can be used to mine telco customer and prospect databases

to gain residential and business customer market share. Four key fuzzy enhancements to traditional database marketing are developed in this paper. First, customers often have significant membership values in more than one distinct fuzzy cluster and can be considered in a natural manner for hybrid or multiple contacts in a given marketing campaign. Second, fuzzy clustering outcomes are shown to be dependent on the particular offer or marketing message. Third, there are differences in clustering outcomes over time, as various offers and treatments are successively presented to consumers, and as products and tastes change. This evolution of fuzzy clusters can be used to help understand customer loyalty and to extract a more optimal lifetime economic relationship value. Fourth, in the longer run, formal procedures can be suggested involving intuitive fuzzy-based clustering metrics for continuous process improvement, to support increasingly flexible and opportunistic campaign management.

Steven Schockaert [19] et.al propser a new Gazetteer services, including geographic search engines and question answering systems. Unfortunately, the footprints provided by gazetteers are often limited to a bounding box or even a centroid. Moreover, for a lot of non–political regions, detailed footprints are nonexistent since theseregions tend to have gradual, rather than crisp, boundaries. In this paper an automatic method to approximate the footprints of crisp, as well as imprecise, regions using statements on the web as a starting Point is proposed. Due to the vague nature of some of these statements, the resulting footprints are represented as fuzzy set.

Kraft, D.H [20] et.al presents an integrated approach to information retrieval which combines fuzzy clustering and fuzzy inference techniques in order to achieve optimal retrieval performance. To capture the relationships among index terms, fuzzy logic rules are used. We adapt several fuzzy clustering methods to the task of clustering documents with respect to the index terms. The clusters generated provide a basis for building the fuzzy logic rules. The clusters can also be used to form hyperlinks between documents. The fuzzy logic rules are applied with fuzzy inference to derive useful modifications of the initial query, which will guide the search for relevant documents. Alternative ways to use the fuzzy clusters are explored in this work as well. Our method combines fuzzy clustering and fuzzy inference with traditional relevance feedback approach for retrieval. The advantage of this approach is the emphasis on semantic information which relates the terms through the fuzzy clusters and fuzzy rules. A series

okyof experiments have been conducted in order to validate this approach; descriptions of those experiments along with the results are presented.

N. Mac Parthalain, R. Jensen and Q. Shen [21] explained that Feature Selection (FS) is a dimensionality reduction technique that aims to select a subset of the original features of a dataset which offer the most useful information. The benefits of feature selection include improved data visualisation, transparency, reduction in training and utilisation times and improved prediction performance. Methods based on fuzzy-rough set theory (FRFS) have employed the dependency function to guide the process with much success. This paper presents a novel fuzzy-rough FS technique which is guided by fuzzy entropy. The use of this measure in fuzzy-rough feature selection can result in smaller subset sizes than those obtained through FRFS alone, with little loss or even an increase in overall classification accuracy.

J.W. Grzymala-Busse and M. Hu [22] proposed that One of the many successful applications of rough set theory has been to this area.The rough set ideology of using only the supplied data and no other information has many benefits in FS, where most other methods require supplementary knowledge. However, the main limitation of rough set-based feature selection in the literature is the restrictive requirement that all data is discrete. In classical rough set theory, it is not possible to consider real-valued or noisy data. This thesis proposes and develops an approach based on fuzzy-rough sets, fuzzy rough feature selection (FRFS), that addresses these problems and retains dataset semantics. Complexity analysis of the underlying algorithms is included.

T-P Hong, L-H Tseng and B-C. Chien [23] described that Over the past years, methods for the automated induction of models and the extraction of interesting patterns from empirical data have attracted considerable attention in the fuzzy set community. This paper briery reviews some typical applications and highlights potential contributions that fuzzy set theory can make to machine learning, data mining, and related ells. The paper concludes with a critical consideration of recent developments and some suggestions for future research directions.

K.M. Faraoun, and A. Boukelif [24] describes a new approach of classification using genetic programming. The proposed technique consists of genetically coevolving a population of non-linear transformations on the input data to be classified, and map them to a new space with a

reduced dimension, in order to get a maximum inter-classes discrimination. The classification of new samples is then performed on the transformed data, and so become much easier. Contrary to the existing GP-classification techniques, the proposed one use a dynamic repartition of the transformed data in separated intervals, the efficacy of a given intervals repartition is handled by the fitness criterion, with a maximum classes discrimination. Experiments were first performed using the Fisher's Iris dataset, and then, the KDD'99 Cup dataset was used to study the intrusion detection and classification problem.

M. Bramrier and W. Banzhaf [25] presented A comparison between four evolutionary techniques for solving symbolic regression problems. The compared methods are multi-expression programming, gene expression programming, grammatical evolution, and linear genetic programming. The comparison includes all aspects of the considered evolutionary algorithms: individual representation, fitness assignment, genetic operators, and evolutionary scheme. Several numerical experiments using five benchmarking problems are carried out. Two test problems are taken from PROBEN1 and contain real-world data. The results reveal that multi-expression programming has the best overall behavior for the considered test problems.

Ralf Mikut, Jens Jäkel, Lutz Gröll [26] presents a method for an automatic and complete design of fuzzy systems from data. The main objective is to build fuzzy systems with a user-controllable trade-off between accuracy and interpretability. Whereas criteria for accuracy mostly follow straightforwardly from the application, definition of interpretability and its criteria are subject to controversial discussion. For this reason, a set of interpretability criteria is given which guide the design process. Consequently, interpretability is maintained by structural choices regarding the type of membership functions, rules, and inference mechanism, on the one hand, and by including interpretability criteria in the rule/rule base evaluation, on the other hand. An application in Instrumented Gait Analysis, to characterize a certain group of patients in comparison to healthy subjects, illustrates the proposed algorithm.

R. Jensen and Q. Shen [27] described that Attribute selection (AS) refers to the problem of selecting those input attributes or features that are most predictive of a given outcome; a problem encountered in many areas such as machine learning, pattern recognition and signal processing.

Unlike other dimensionality reduction methods, attribute selectors preserve the original meaning of the attributes after reduction. This has found application in tasks that involve datasets containing huge numbers of attributes (in the order of tens of thousands) which, for some learning algorithms, might be impossible to process further. Recent examples include text processing and web content classification. AS techniques have also been applied to small and medium-sized datasets in order to locate the most informative attributes for later use. One of the many successful applications of rough set theory has been to this area. The rough set ideology of using only the supplied data and no other information has many benefits in AS, where most other methods require supplementary knowledge. However, the main limitation of rough set-based attribute selection in the literature is the restrictive requirement that all data is discrete. In classical rough set theory, it is not possible to consider real-valued or noisy data. This paper investigates a novel approach based on fuzzy-rough sets, fuzzy rough feature selection (FRFS), that addresses these problems and retains dataset semantics. FRFS is applied to two challenging domains where a feature reducing step is important; namely, web content classification and complex systems monitoring. The utility of this approach is demonstrated and is compared empirically with several dimensionality reducers. In the experimental studies, FRFS is shown to equal or improve classification accuracy when compared to the results from unreduced data. Classifiers that use a lower dimensional set of attributes which are retained by fuzzy-rough reduction outperform those that employ more attributes returned by the existing crisp rough reduction method. In addition, it is shown that FRFS is more powerful than the other AS techniques in the comparative study.

Włodzisław Duch [28] presented Similarity-based methods (SBM), those are a generalization of the minimal distance (MD) methods which form a basis of several machine learning and pattern recognition methods. Investigation of similarity leads to a fruitful framework in which many classification, approximation and association methods are accommodated. Probability $p(C|\mathbf{X};M)$ of assigning class C to a vector $\mathbf{X}$, given a classification model M, depends on adaptive parameters and procedures used in construction of the model. Systematic overview of choices available for model building is described and numerous improvements suggested. Similarity-Based Methods have natural neural-network type realizations. Such neural network models as the Radial Basis Functions (RBF) and the Multilayer Perceptions (MLPs) are included in this

framework as special cases. SBM may also include several different sub models and a procedure to combine their results. Many new versions of similarity-based methods are derived from this framework. A search in the space of all methods belonging to the SBM framework finds a particular combination of parameterizations and procedures that is most appropriate for a given data. No single classification method can beat this approach. Preliminary implementation of SBM elements tested on a real world datasets gave very good results.

Richard E. Haskell [29] explained that the terminal nodes of a binary tree classifier represent discrete classes to be recognized. In this paper the classes are considered to be fuzzy sets in which a specific sample can belong to more than one class with different degrees of membership. The terminal nodes is this case will contain information about the degrees to which test samples belong to particular classes. This will allow the development of a regression tree in which a continuous output value such as the control signal of a fuzzy controller can be learned. In addition to the classes being fuzzy sets each node of the regression tree is made fuzzy by associating a membership function with the fuzzy sets feature_value < threshold and feature_value >= threshold. The output value is found by dropping the input measurement vector through the tree in which it will, in general, take both paths at each node with a weighting factor determined by the node membership functions. The crisp output value (defuzzification) is a weighted sum of the class values associated with the terminal nodes. The splitting criterion for each tree node is based on the use of a fuzzy cumulative distribution function, which is a generalization of the Kolmogorov-Smirnov (K-S) distance suitable for multiple classes. The splitting of nodes is terminated when all training samples belonging to a given node have their maximum degree of membership associated with a given class. Large decision trees are typically pruned to provide better classification accuracy when used with test data. A stock market prediction example is used to show that making a large fuzzy tree is an attractive alternative to pruning. Fuzzy classification and regression trees can be considered to be a fuzzy neural network in which the structure of the network is learned rather than the weights. Such neuro-fuzzy classification and regression trees should lend themselves to efficient implementation in a VLSI chip in which each test sample can propagate through all paths simultaneously.

INTRODUCTION

In recent years machine learning and knowledge discovery techniques have attracted a great deal of attention in the information area. Classification is one of the important research topics on these research areas. Most of researches on classification concern that a complete data set is given as a training set and the test data know all values of attributes clearly. Unfortunately, incomplete data are commonly seen in real world applications. Knowledge discovery algorithms take an input of training examples of target knowledge, and output a fuzzy logic formula that best fits the training examples. The execution is done in some steps and it could be made possible by using object, data input, algorithm, process, experiment, and results.

Fuzzy Logic is a form of logic that extends on Boolean logic that incorporates partial values of truth - Instead of sentences being "Completely true" or "Completely false," they are assigned a value that represents their degree of truth. In fuzzy systems, values are indicated by a number (called a truth value) in the range from 0 to 1, where 0.0 represents absolute falseness and 1.0 represents absolute truth. Fuzzification is the generalization of any theory from discrete to continuous. Fuzzy Logic is important to AI because they allow computers to answer 'to a certain degree' as opposed to in one extreme or the other. In this sense, computers are allowed to think more 'human-like' since almost nothing in our perception is extreme, but is true only to a certain degree. Through fuzzy logic, machines can think in degrees, solve problems when there is no simple mathematical model, solve problems for highly nonlinear processes and use expert knowledge to make decisions.

Knowledge discovery in Fuzzy logic is based on membership function values. After a fundamental algorithm, fuzzy logic functions are applied to a more practical example of classification problem, in which expressiveness of fuzzy logic functions is examined for a well-known machine-learning database. Here in this work, we have investigated the problem of incomplete data in data sets in the input-output behavior of a Synchronous Generator. A data set with at least one missing attribute value is referred as an incomplete data set.

Since the incomplete samples don't provide perfect information for training process, most of the traditional classification algorithms cannot be with incomplete data directly but generate in accurate classifiers from an incomplete data. Hence the incomplete data must be tackled well so that good classification models can be developed for real life applications. The genetic programming is one of the techniques on evolutionary computation. The genetic programming has been applied to several applications like symbolic regression, the robot control programs, and classification, etc. genetic programming can discover underlying data relationships and present these relationships by expression. A supervised learning method based on genetic programming to handle the classification problem with incomplete data in attributes has been used.

In this thesis a new strategy based on Fuzzy Entropy has been introduced for the first time to deal with the incomplete quantitative data in the case of Synchronous Generator. For handling incomplete quantitative data, we have firstly applied fuzzy entropy to discriminate the best number of intervals, which have been granulated as a fuzzy linguistic term with a membership function. Then, we employ the linguistic term to infer the missing attribute values based on the max-min composition method according to their class labels. This paper also introduces a supervised learning method based on genetic programming to handle the classification problem with incomplete data in attributes.

CHAPTER-1 DATA AND INFORMATION

1.1 INTRODUCTION:

In a broad sense, a **data** defines a set of values, and the allowable operations on those values. In programming languages a data type is an attribute of a piece of data that tells the computer (and the programmer) something about what kind of data is being dealt with. This involves setting constraints on the data, such as what values that data can take on, and what operations may be performed on that data. Common data types may include: integers, floating-point numbers (decimals), and alphanumeric strings. For example, in the Java programming language, the "int" type represents the set of 32-bit integers ranging in value from -2,147,483,648 to 2,147,483,647, as well as the operations that can be performed on integers, such as addition, subtraction, and multiplication. Colors, on the other hand, are represented by three bytes denoting the amounts each of red, green, and blue, and one string representing that color's name; allowable operations include addition and subtraction, but not multiplication.

Almost all programming languages explicitly include the notion of data type, though different languages may use different terminology. Most programming languages also allow the programmer to define additional data types, usually by combining multiple elements of other types and defining the valid operations of the new data type. For example, a programmer might create a new data type named "Person" that specifies that data interpreted as Person would include a name and a date of birth.

A data type can also be thought of as a constraint placed upon the interpretation of data in a type system, describing representation, interpretation and structure of values or objects stored in computer memory. The type system uses data type information to check correctness of computer programs that access or manipulate the data.

1. Machine data types
2. Primitive types
3. Composite types
4. Abstract data types

5. Pointer and reference types
6. Algebraic types
7. Object types
8. Function types

In programming, classification of a particular type of information. It is easy for humans to distinguish between different types of data. We can usually tell at a glance whether a number is a percentage, a time, or an amount of money. We do this through special symbols -- %: and $ -- that indicate the data's type. Similarly, a computer uses special internal codes to keep track of the different types of data it processes.

Most programming languages require the programmer to declare the data type of every data object, and most database systems require the user to specify the type of each data field. The available data types vary from one programming language to another, and from one data base application to another, but the following usually exist in one form or another:

Integer **:** In more common parlance, whole number; a number that has no fractional part.

Floating-Point **:** A number with a decimal point. For example, 3 is an integer, but 3.5 is a floating-point number.

We'll talk about data in lots of places in The Knowledge Base, but here I just want to make a fundamental distinction between two types of data: **qualitative** and **quantitative**. The way we typically define them, we call data 'quantitative' if it is in numerical form and 'qualitative' if it is not. Notice that qualitative data could be much more than just words or text. Photographs, videos, sound recordings and so on, can be considered qualitative data.

Personally, while I find the distinction between qualitative and quantitative data to have some utility, I think most people draw too hard a distinction, and that can lead to all sorts of confusion. In some areas of social research, the qualitative-quantitative distinction has led to protracted arguments with the proponents of each arguing the superiority of their kind of data over the other. The quantitative types argue that their data is 'hard', 'rigorous', 'credible', and 'scientific'. The qualitative proponents counter that their data is 'sensitive', 'nuanced', 'detailed', and 'contextual'.

For many of us in social research, this kind of polarized debate has become less than productive. And, it obscures the fact that qualitative and quantitative data are intimately related to each other. **All quantitative data is based upon qualitative judgments; and all qualitative data can be described and manipulated numerically.** For instance, think about a very common quantitative measure in social research -- a self esteem scale. The researchers who develop such instruments had to make countless judgments in constructing them: how to define self esteem; how to distinguish it from other related concepts; how to word potential scale items; how to make sure the items would be understandable to the intended respondents; what kinds of contexts it could be used in; what kinds of cultural and language constraints might be present; and on and on. The researcher who decides to use such a scale in their study has to make another set of judgments: how well does the scale measure the intended concept; how reliable or consistent is it; how appropriate is it for the research context and intended respondents; and on and on. Believe it or not, even the respondents make many judgments when filling out such a scale: what is meant by various terms and phrases; why is the researcher giving this scale to them; how much energy and effort do they want to expend to complete it, and so on. Even the consumers and readers of the research will make lots of judgments about the self esteem measure and its appropriateness in that research context. What may look like a simple, straightforward, cut-and-dried quantitative measure is actually based on lots of qualitative judgments made by lots of different people.

On the other hand, all qualitative information can be easily converted into quantitative, and there are many times when doing so would add considerable value to your research. The simplest way to do this is to divide the qualitative information into units and number them! I know that sounds trivial, but even that simple nominal enumeration can enable you to organize and process qualitative information more efficiently. Perhaps more to the point, we might take text information (say, excerpts from transcripts) and pile these excerpts into piles of similar statements. When we do something even as easy as this simple grouping or piling task, we can describe the results quantitatively. For instance, if we had ten statements and we grouped these into five piles (as shown in the figure).

We could describe the piles using a 10 x 10 table of **0**'s and **1**'s. If two statements were placed together in the same pile, we would put a **1** in their row-column juncture. If two statements were placed in different piles, we would use a **0**. The resulting matrix or table describes the grouping of the ten statements in terms of their similarity. Even though the data

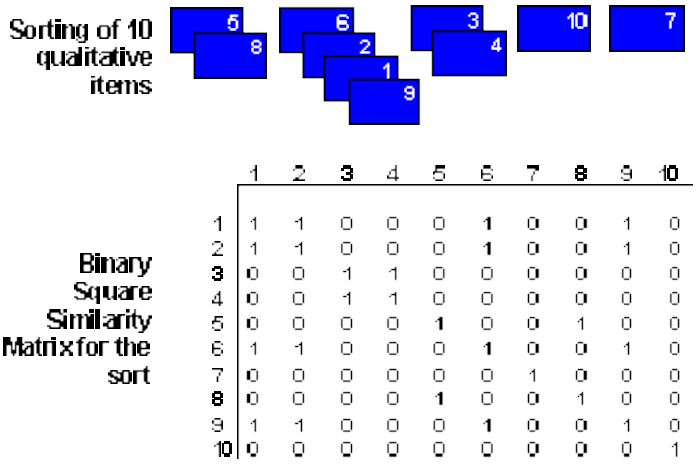


Figure: 1.1 Qualitative Information

in this example consists of qualitative statements (one per card), the result of our simple qualitative procedure (grouping similar excerpts into the same piles) is ***quantitative*** in nature. "So what?" you ask. Once we have the data in numerical form, we can manipulate it numerically. For instance, we could have five different judges sort the 10 excerpts and obtain a 0-1 matrix like this for each judge. Then we could average the five matrices into a single one that shows the proportions of judges who grouped each pair together. This proportion could be considered an estimate of the similarity (across independent judges) of the excerpts. While this might not seem too exciting or useful, it is exactly this kind of procedure that I use as an integral part of the process of developing 'concept maps' of ideas for groups of people (something that ***is*** useful!).

A dictionary defines data as facts or figures from which conclusions may be drawn. Thus, technically, it is a collective or plural noun. Some recent dictionaries acknowledge popular usage of the word data with a singular verb. However we intend to adhere to the traditional "English" teacher mentality in our grammar usage—sorry if "data are" just doesn't sound quite right! (My mother and step-mother were both English teachers, so clearly no offense is intended above.) Datum is the singular form of the noun data. Data can be classified as either numeric or nonnumeric. Specific terms are used as follows:

1. {Poor, Fair, Good, Better, Best}, colors (ignoring any physical causes), and types of material {straw, sticks, bricks} are examples of qualitative data.
2. Qualitative data are often termed categorical data. Some books use the terms individual and variable to reference the objects and characteristics described by a set of data.

3. They also stress the importance of exact definitions of these variables, including what units they are recorded in. The reason the data were collected is also important.
4. Quantitative data are further classified as either discrete or continuous.
5. A classic example of discrete data is a finite subset of the counting numbers, {1, 2, 3, 4, and 5} perhaps corresponding to {strongly Disagree... Strongly Agree}.

Another classic is the electric charge of a single electron which was first convincingly measured in 1911 in the Millikan Oil-drop Experiment. Quantum Mechanics, the field of physics which deals with the very small, is much concerned with discrete values. When data represent counts, they are discrete. An example might be how many students were absent on a given day. Counts are usually considered exact and integer. Consider, however, if three tradies make an absence, then aren't two tar dies equal to 0.67 absences? The real numbers are continuous with no gaps or interruptions. Physically measureable quantities of length, volume, time, mass, *etc.* are generally considered continuous. At the physical level (microscopically), especially for mass, this may not be true, but for normal life situations is a valid assumption.

The structure and nature of data will greatly affect our choice of analysis method. By structure we are referring to the fact that, for example, the data might be pairs of measurements. Consider the legend of Galileo dropping weights from the leaning tower of Pisa. The times for each item would be paired with the mass (and surface area) of the item. Something which Galileo clearly did was measure the time it took a pendulum to swing with various amplitudes.

1.2 LEVELS OF MEASUREMENT:

The experimental (scientific) method depends on physically measuring things. The concept of measurement has been developed in conjunction with the concepts of numbers and units of measurement. Statisticians categorize measurements according to levels. Each level corresponds to how this measurement can be treated mathematically.

1. **Nominal**: Nominal data have no order and thus only gives **names** or labels to various categories.
2. **Ordinal**: Ordinal data have **order**, but the interval between measurements is not meaningful.

3. **Interval**: Interval data have meaningful intervals between measurements, but there is no true starting point (zero).
4. **Ratio**: Ratio data have the highest level of measurement. Ratios between measurements as well as intervals are meaningful because there is a starting point (zero).

Nominal comes from the Latin root nomen meaning name. Nomenclature, nominative, and nominee are related words. Gender is nominal. (Gender is something you are born with, whereas **sex** is something you should get a license for.)

Example1:Colors

To most people, the colors: black, brown, red, orange, yellow, green, blue, violet, gray, and white are just names of colors. To an electronics student familiar with color-coded resistors, this data is in ascending order and thus represents at least ordinal data. To a physicist, the colors: red, orange, yellow, green, blue, and violet correspond to specific wavelengths of light and would be an example of ratio data.

Example2**:**Temperatures

What level of measurement a temperature is depends on which temperature scale is used. Specific values: 0°C = 32°F = 273.15 K = 491.69°R 100°C = 212°F = 373.15 K = 671.67°R -17.8°C = 0°F = 255.4 K = 459.67°R where C refers to Celsius (or Centigrade before 1948); F refers to Fahrenheit; K refers to Kelvin; R refers to Rankine. Only Kelvin and Rankine have true zeroes (starting point) and ratios can be found. Celsius and Fahrenheit are interval data; certainly order is important and intervals are meaningful. However, a 180° dashboard is not twice as hot as the 90° outside temperature (Fahrenheit assumed)! Rankine has the same size degree as Fahrenheit but is rarely used. To interconvert Fahrenheit and Celsius, see Numbers lesson 12. (Note that since 1967, the use of the degree symbol on tempertures Kelvin is no longer proper.) Although ordinal data should not be used for calculations, it is not uncommon to find averages formed from data collected which represented Strongly Disagree, Strongly Agree! Also, averages of nominal data (zip codes, social security numbers) are rather meaningless!

The Java programming language is strongly-typed, which means that all variables must first be declared before they can be used. This involves stating the variable's type and name, as you've already seen: int gear = 1;

Doing so tells your program that a field named "gear" exists, holds numerical data, and has an initial value of "1". A variable's data type determines the values it may contain, plus the operations that may be performed on it. In addition to `int`, the Java programming language supports seven other primitive data types. A primitive type is predefined by the language and is named by a reserved keyword. Primitive values do not share state with other primitive values. The eight primitive data types supported by the Java programming language are:

- **Byte**: The `byte` data type is an 8-bit signed two's complement integer. It has a minimum value of -128 and a maximum value of 127 (inclusive). The `byte` data type can be useful for saving memory in large arrays, where the memory savings actually matters. They can also be used in place of `int` where their limits help to clarify your code; the fact that a variable's range is limited can serve as a form of documentation.
- **Short**: The `short` data type is a 16-bit signed two's complement integer. It has a minimum value of -32,768 and a maximum value of 32,767 (inclusive). As with `byte`, the same guidelines apply: you can use a `short` to save memory in large arrays, in situations where the memory savings actually matters.
- **Int**: The `int` data type is a 32-bit signed two's complement integer. It has a minimum value of -2,147,483,648 and a maximum value of 2,147,483,647 (inclusive). For integral values, this data type is generally the default choice unless there is a reason (like the above) to choose something else. This data type will most likely be large enough for the numbers your program will use, but if you need a wider range of values, use `long` instead.
- **Long**: The `long` data type is a 64-bit signed two's complement integer. It has a minimum value of -9,223,372,036,854,775,808 and a maximum value of 9,223,372,036,854,775,807 (inclusive). Use this data type when you need a range of values wider than those provided by `int`.

- **Float**: The `float` data type is a single-precision 32-bit IEEE 754 floating point. Its range of values is beyond the scope of this discussion, but is specified in section 4.2.3 of the Java Language Specification. As with the recommendations for `byte` and `short`, use a `float` (instead of `double`) if you need to save memory in large arrays of floating point numbers. This data type should never be used for precise values, such as currency. For that, you will need to use the java. Math. Big Decimal class instead. Numbers and Strings covers `Big Decimal` and other useful classes provided by the Java platform.
- **Double**: The `double` data type is a double-precision 64-bit IEEE 754 floating point. Its range of values is beyond the scope of this discussion, but is specified in section 4.2.3 of the Java Language Specification. For decimal values, this data type is generally the default choice. As mentioned above, this data type should never be used for precise values, such as currency.
- **Boolean**: The `Boolean` data type has only two possible values: `true` and `false`. Use this data type for simple flags that track true/false conditions. This data type represents one bit of information, but its "size" isn't something that's precisely defined.
- **Char**: The `char` data type is a single 16-bit Unicode character. It has a minimum value of `'\u0000'` (or 0) and a maximum value of `'\uffff'` (or 65,535 inclusive).

In addition to the eight primitive data types listed above, the Java programming language also provides special support for character strings via the java.lang.String class. Enclosing your character string within double quotes will automatically create a new `String` object; for example, `String s = "this is a string";`. `String` objects are *immutable*, which means that once created, their values cannot be changed. The `String` class is not technically a primitive data type, but considering the special support given to it by the language, you'll probably tend to think of it as such. A data type in a programming language is a set of data with values having predefined characteristics. Examples of data types are: integer, floating point unit number, character, string, and pointer. Usually, a limited number of such data types come built into a language. The language usually specifies the range of values for a given data type, how the values are processed by the computer, and how they are stored. With object-oriented programming, a programmer can create new data types to meet application needs. Such an exercise as known as "data abstraction" and the result is a new class of data. Such a class can

draw upon the "built-in" data types such as number integers and characters. For example, a class could be created that would abstract the characteristics of a purchase order. The purchase order data type would contain the more basic data types of numbers and characters and could also include other object defined by another class. The purchase order data type would have all of the inherent services that a programming language provided to its built-in data types.

1.3 NUMERICAL DATA (OR QUANTITATIVE DATA):

It is data measured or identified on a numerical scale. Numerical data can be analyzed using statistical methods, and results can be displayed using tables, charts, histograms and graphs. For example, a researcher will ask a questions to a participant that include words how often, how many or percentage. The answers from the questions will be numerical.

Mathematica provides integrated support both for classical statistics and for modern large-scale data analysis. Its symbolic character allows broader coverage, with symbolic manipulation of statistical distributions, symbolic specification of functions and models, and general symbolic representations of large-scale data properties. Incorporating the latest numerics and computational geometry algorithms, Mathematica provides high-accuracy and high-reliability statistical results for datasets of almost unlimited size.

1.3.1 MEAN:

Means of random reals approach a Gaussian distribution:

```
In[1]:= ListPlot[Sort[Table[Mean[RandomReal[1, 20]], {1000}]]]
```

Out[1]=

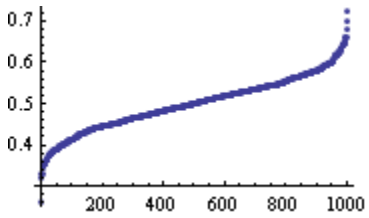


Figure: 1.2 Mean Of Random Distribution

Theoretical curve:

```
In[2]:= Plot[InverseCDF[NormalDistribution[1/2, Sqrt[1/12]/Sqrt[20]], q/1000],
         {q, 1, 1000}]
```

Out[2]=

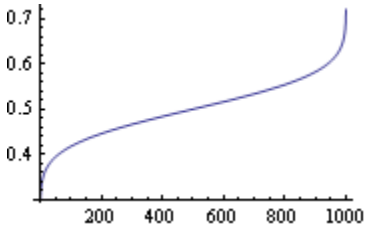


Figure 1.3: Theoretical Curve

Successive averages (Cesàro summation):

```
In[1]:= Table[Array[a, k], {k, 4}]

Out[1]= {{a[1]}, {a[1], a[2]}, {a[1], a[2], a[3]}, {a[1], a[2], a[3], a[4]}}

In[2]:= Mean /@ %

Out[2]= {a[1], 1/2 (a[1] + a[2]), 1/3 (a[1] + a[2] + a[3]), 1/4 (a[1] + a[2] + a[3] + a[4])}
```

Mean values of cells in a sequence of steps of 2D cellular automaton evolution:

```
In[1]:= ArrayPlot[Mean[CellularAutomaton[{14, {2, 1}, {1, 1}}, {{{1}}, 0}, 30]]]
```

Out[1]=

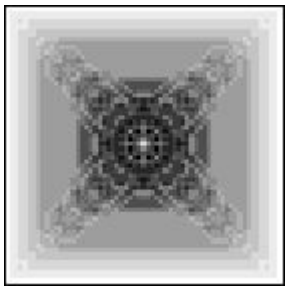


Figure: 1.4 mean of cells

1.3.2 MEDIAN:

Obtain a robust estimate of location when outliers are present:

```
In[1]:= Median[{1, 5, 2, 6, 10, 10^5, 5, 4}]

Out[1]= 5
```

Extreme values have a large influence on the Mean:

In[2]:=
```
Mean[{1, 5, 2, 6, 10, 10^5, 5, 4}] // N
```

Out[2]=
```
12504.1
```

Medians of random real approach a Gaussian distribution:

In[1]:=
```
ListPlot[Sort[Table[Median[RandomReal[1, 100]], {1000}]]]
```

Out[1]=

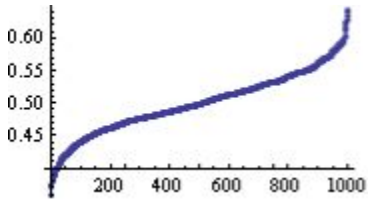


Figure 1.5: Medians of Random Distribution

Theoretical curve:

In[2]:=
```
sd = 1 / PDF[UniformDistribution[{0, 1}], 1 / 2] / 2 / Sqrt[100]
```

Out[2]=
$$\frac{1}{20}$$

In[3]:=
```
Plot[InverseCDF[NormalDistribution[1 / 2, sd], q / 1000], {q, 1, 1000}]
```

Out[3]=

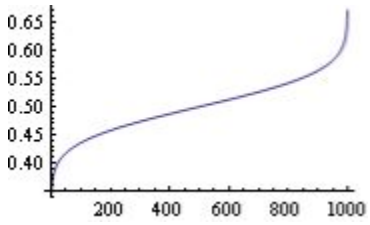


Figure 1.6: Theoretical Curve (median)

1.3.3 LEAST SQUARES:

Here is some data:

In[1]:=
```
x = RandomReal[10, 1000];
y = Sin[x] RandomReal[1, 1000];
data = Transpose[{x, y}];
ListPlot[data]
```

Out[1]=

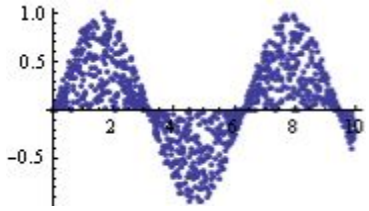


Figure: 1.7 Least Square

Define cubic basis functions centered at t with support on the interval [t - 2, t + 2]:

In[2]:=
```
b[t_][r_] =
  Block[{s = r - t, sp1}, Piecewise[{{0, s ≤ -2}, {(2 + s)^3, -2 < s ≤ -1},
     {1 + 3 (1+s) (1 - s (1+s)), -1 < s < 0}, {1 + 3 (1-s) (1 + s*(1-s)),
      0 <= s < 1}, {(2 - s)^3, 1 ≤ s < 2}, {0, s ≥ 2}}]];
```

In[3]:=
```
Plot[b[0][x], {x, -2, 2}]
```

Out[3]=

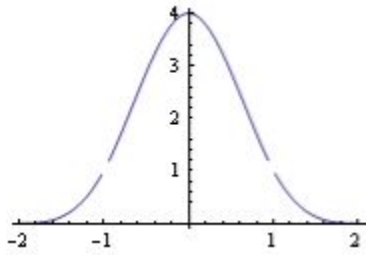


Figure: 1.8 Distribution Curve

Set up a sparse design matrix for basis functions centered at 0, 1, ..., 10:

In[4]:=
```
s = SparseArray[Flatten[
   Table[Table[{i, t + 1} -> b[t][x[[i]]], {t, Max[0, Floor[x[[i]] - 2]],
      Min[10, Ceiling[x[[i]] + 2]]}], {i, Length[x]}]]]
```

Out[4]=
```
SparseArray[<3802>, {1000, 11}]
```

Solve the least-squares problem:

In[5]:=
```
c = LeastSquares[s, y]
```

Out[5]=
```
{-0.015761, 0.0961337, 0.0877775, 0.01304, -0.0793905,
 -0.0923857, -0.0284206, 0.0703658, 0.0987393, 0.0466853, -0.0673069}
```

In[6]:=
```
Show[ListPlot[data],
 Plot[c.Table[b[t][x], {t, 0, 10}], {x, 0, 10}, PlotStyle → Red]]
```

Out[6]=

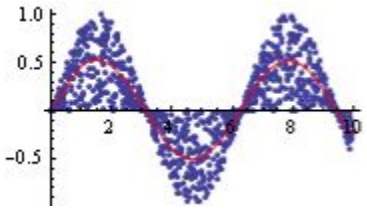


Figure: 1.9 Output Curve Using Least Square

1.3.4 STANDARD DEVIATION:

Transform data to have mean 0 and unit variance:

In[1]:=
```
data = RandomReal[5, 20]
```

Out[1]=
```
{1.4853, 0.699014, 1.18438, 4.93687, 3.04651, 2.32552,
 3.3561, 4.68015, 4.56495, 3.84699, 4.54357, 1.15567, 0.698955,
 0.737586, 0.688725, 1.35113, 4.12072, 2.85536, 1.91749, 0.220822}
```

In[2]:=
```
standardized = (data - Mean[data]) / StandardDeviation[data]
```

Out[2]=
```
{-0.582942, -1.0729, -0.770451, 1.56786, 0.389908, -0.0593639,
 0.582827, 1.40789, 1.3361, 0.888721, 1.32278, -0.788345, -1.07294,
 -1.04887, -1.07931, -0.666546, 1.05929, 0.270795, -0.313625, -1.37088}
```

In[3]:=
```
Mean[standardized] // Chop
```

Out[3]=
```
0
```

In[4]:=
```
Variance[standardized]
```

Out[4]=
```
1.
```

1.3.5 COVARIANCE:

Compute the covariance of two financial time series:

In[1]:=
```
Covariance[FinancialData["^GSPC", "Price", {2004, 1, 1}, "Value"],
 FinancialData["^DJI", "Price", {2004, 1, 1}, "Value"]]
```

Out[1]= 63161.7

1.3.6 NORMAL DISTRIBUTION:

Compute p-values for a z-test with alternative hypothesis $X < z$:

In[1]:=
```
CDF[NormalDistribution[], z]
```

Out[1]= $\frac{1}{2}\left(1+\text{Erf}\left[\frac{z}{\sqrt{2}}\right]\right)$

Alternative hypothesis $X > z$:

In[2]:=
```
1 - CDF[NormalDistribution[], z]
```

Out[2]= $1+\frac{1}{2}\left(-1-\text{Erf}\left[\frac{z}{\sqrt{2}}\right]\right)$

Alternative hypothesis $|X| > z$:

In[3]:=
```
2 CDF[NormalDistribution[], z]
```

Out[3]= $1+\text{Erf}\left[\frac{z}{\sqrt{2}}\right]$

Plot the cumulative distribution function of the random variable:

In[1]:=
```
Plot[CDF[NormalDistribution[0, 1], x], {x, -3, 3}]
```

Out[1]=

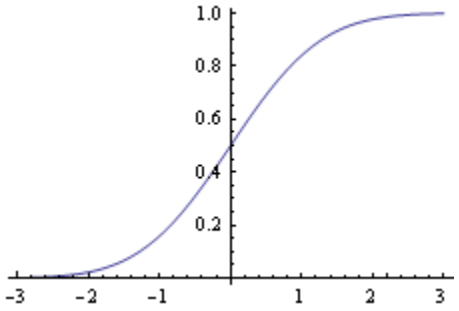


Figure: 1.10 Normal Distribution

A contour plot as both $x$and $\sigma$are varied:

In[1]:=
```
ContourPlot[CDF[NormalDistribution[0, σ], x], {x, -3, 3}, {σ, 1/10, 5}]
```

Out[1]=

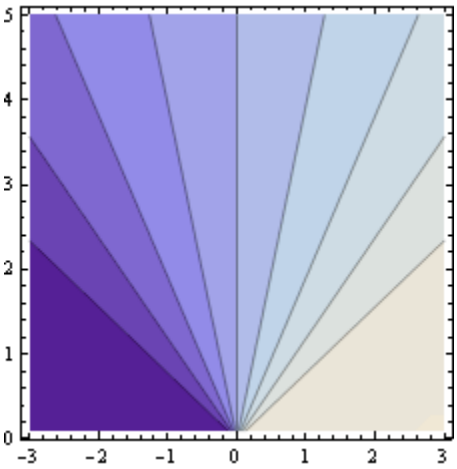


Figure: 1.11 Normal Distribution In Plane

In[1]:=
```
Graphics[Point[RandomReal[NormalDistribution[], {2000, 2}]]]
```

Out[1]=

Figure: 1.12 Normally distributed points in 3D

In[1]:=
```
Graphics3D[Point[RandomReal[NormalDistribution[], {5000, 3}]]]
```

1.3.7 RANDOM REAL FUNCTION:

A random walk:

In[1]:=
```
ListLinePlot[Accumulate[RandomReal[{-1, 1}, 100]]]
```

Out[1]=

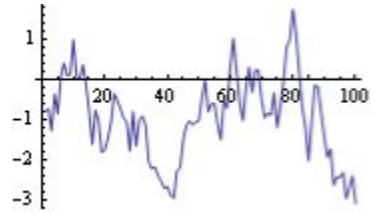


Figure: 1.13 Random Real Function

In[1]:=
```
Graphics[Circle /@ RandomReal[10, {40, 2}]]
```

Out[1]=

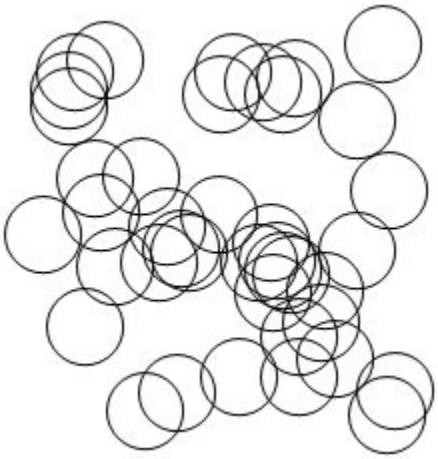


figure: 1.14 Circles At Random Positions

In[1]:=
```
ArrayPlot[RandomReal[1, {30, 40}]]
```

Out[1]=

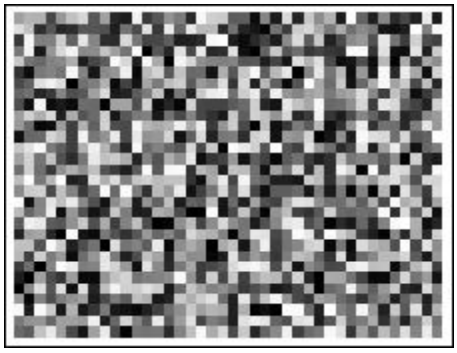


Figure: 1.15 Random Array of Gray Levels:

In[1]:= `Graphics3D[Sphere /@ RandomReal[10, {50, 3}]]`

Out[1]=

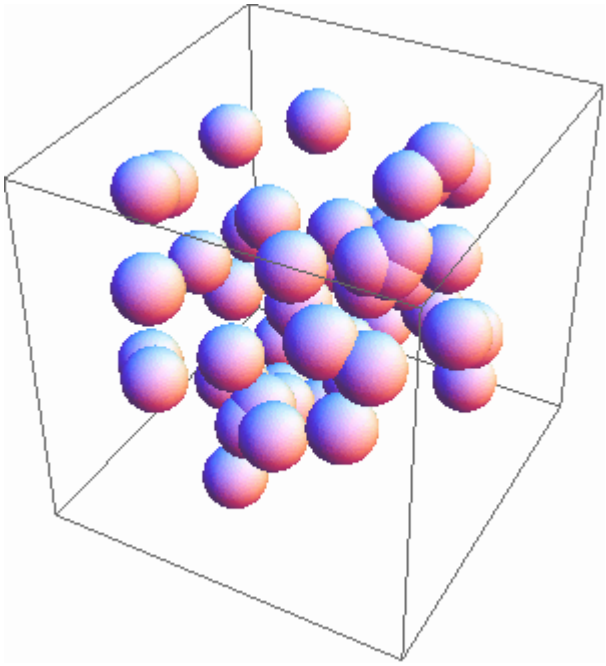


Figure: 1.16 Spheres At Random Positions:

2D random walk:

In[1]:= `Graphics[Line[Accumulate[RandomReal[{-1, 1}, {500, 2}]]]]`

Out[1]=

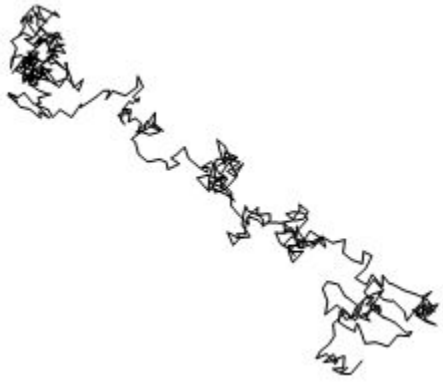


Figure: 1.17 2D Random Walk:

3D random walk:

```
In[2]:= Graphics3D[Line[Accumulate[RandomReal[{-1, 1}, {500, 3}]]]]
```

Out[2]=

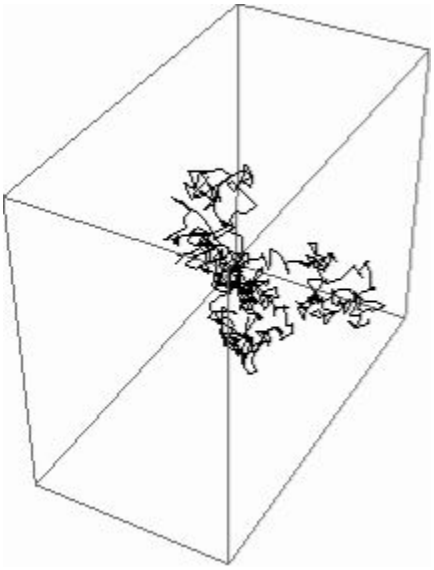


Figure: 1.18 3D Random Walk:

Determinants of random 100×100 matrices:

```
In[1]:= Table[Det[RandomReal[1, {100, 100}]], {10}]

Out[1]= {-1.56089×10^25, -2.63437×10^24, -3.27404×10^25, -1.1467×10^25, -2.06694×10^25,
  1.10247×10^24, 2.19787×10^24, 3.03921×10^25, -3.02607×10^25, 2.49147×10^24}
```

Generate a complex number in the unit square:

```
In[1]:= Complex@@ RandomReal[1, 2]

Out[1]= 0.491045 + 0.334147 i
```

Generate 5 complex numbers:

```
In[2]:= Complex@@@ RandomReal[1, {5, 2}]

Out[2]= {0.659002 + 0.558844 i, 0.176329 + 0.550189 i,
  0.572141 + 0.924842 i, 0.720507 + 0.831983 i, 0.141602 + 0.413622 i}
```

1.4 DATA ACCURACY:

This article covers a definition of data accuracy. Data accuracy is the foundation dimension of data quality. If the data is wrong, the other dimensions matter little. Subsequent articles will cover how to organize a data quality assurance function to focus on data accuracy and a methodological approach, called data profiling, which is used to assess the accuracy component of data quality. Data quality has multiple dimensions. This is my list:

- Accuracy
- Timeliness
- Relevance
- Completeness
- Understood by users
- Trusted by users

Accuracy refers to whether the data correctly records the business object or event it represents. It has two requirements: it must be the right value and it must represent the value in a consistent form with all other representations of the same value. Having the right value means more than being a valid value. Meta data should provide a definition of what values are valid for each field. For example, it can provide a data type, length restrictions, range of acceptable values, discrete list of acceptable values, rules for entering not known or not applicable and other rules such as uniqueness or consecutiveness. These definitions restrict the values to only those that are valid for the field. However, a value can be valid and wrong. For example, if a field named EYE_COLOR contains a value of FORD, it is invalid since it is not a valid color. If it contains a value of BROWN, it is valid. However, to be accurate, the person it refers to must have brown eyes; otherwise it is valid but inaccurate. This demonstrates a limitation of using meta data to find occurrences of inaccurate data. It can be used to only find invalid data, not all inaccurate data. The other component of accuracy touches on representation consistency. For example, a text field containing the name of a state may have entries for TX, TEXAS and Texas. To someone looking at a report, this presents no problem. All entries unambiguously mean the state of Texas. So, they are right values.

The reason these are considered inaccurate is that they will provide inaccurate results from general queries that select on a value or that aggregate on the field. Values need to be consistent in representation in order to provide accurate ad hoc query results. The operational application may not care about consistency. However, decision support, data mining and integration applications all require consistency. For a specific point of interest in three-dimensional space, accuracy is the difference between the actual position in space and the position as measured by a measurement device. Stage accuracy is influenced by the feedback mechanism (linear encoder, rotary encoder, laser interferometer), drive mechanism (ball screw, lead screw, linear motor), and trueness of bearing ways. The measurement reference for Aerotech linear products is a laser interferometer.

1.5 REPEATABILITY:

Repeatability is the variation in measurements obtained when one person takes multiple measurements using the same instrument and techniques on the same parts or items. Repeatability is the variation of outcomes of an experiment carried out in the same conditions, e.g. by the same operator, in the same laboratory. For example, repeatability of measurements of precise mechanical scales is the variation of weight values reported for a given constant mass by the same person, in conditions with the same temperature and humidity. For continuous outcome variables, repeatability may be quantified via such measures as "repeatability standard deviation " or "repeatability variance ". Repeatability is often considered in conjunction with reproducibility_; there is even a common acronym for this pair - "R&R". The closeness of agreement between independent results obtained with the same method on identical test material, under the same conditions (same operator, same apparatus, same laboratory and after short intervals of time). The measure of repeatability is the standard deviation qualified with the term: ‘repeatability’ as repeatability standard deviation. In some contexts repeatability may be defined as the value below which the absolute difference between two single test results obtained under the above conditions, may be expected to lie with a specified probability.

Repeatability is defined as the range of positions attained when the system is repeatedly commanded to one location under identical conditions. Uni-directional repeatability is measured by approaching the point from one direction, and ignores the effects of backlash or hysteresis within the system. Bi-directional repeatability measures the ability to return to the point from

both directions. Many vendors specify repeatability as ± (resolution). This is the repeatability of any digital servo system as measured at the encoder. All of Aerotech's specifications, which include the effects of Abbe error, friction, etc. are based on actual operating conditions and usage not on theoretical, unachievable values.

Gauge repeatability and reproducibility (R&R) studies have been widely used for assessing the precision of a measurement system and identifying its sources of variability, including the contributions from operator, instrument, and random effects. The variation observed when 1 operator measures the same part is called repeatability or pure error. The variation observed when 1 operator duplicates the measurement of another operator on the same part is called reproducibility.

In this study, gauge R&R analysis was performed on the paddle dissolution test measurement system (US Pharmacopeia [USP] apparatus 2)[14] to examine variability from the apparatus, operator, and tablets. JMP (version 5.1) statistical software was used to analyze variability contributions. Based on the analysis of variability sources, a change in the mechanical calibration for the measurement system was made to reduce instrument variability. A subsequent statistical analysis showed that the new mechanical calibration process improved the performance of the dissolution measurement system. This software is capable of determining repeatability and reproducibility using, below method.

- The Range Method,
- The Range Average Method, and
- The ANOVA method

1.6 REPRODUCIBILITY:

Reproducibility is one of the main principles of the scientific method, and refers to the ability of a test or experiment to be accurately reproduced, or replicated, by someone else working independently. Reproducibility is different from repeatability, which measures the success rate in successive experiments, possibly conducted by the same experimenters. Reproducibility relates to the agreement of test results with different operators, test apparatus, and laboratory locations. It is often reported as a standard deviation. While repeatability of scientific experiments is desirable, it is not considered necessary to establish the scientific validity of a theory. For

example, the cloning of animals is difficult to repeat, but has been reproduced by various teams working independently, and is a well established research domain. One failed cloning does not mean that the theory is wrong or unscientific. Repeatability is often low in protosciences.

The closeness of agreement between independent results obtained with the same method on identical test material but under different conditions (different operators, different apparatus, different laboratories and/or after different intervals of time). The measure of reproducibility is the standard deviation qualified with the term ‘reproducibility’ as reproducibility standard deviation. In some contexts reproducibility may be defined as the value below which the absolute difference between two single test results on identical material obtained under the above conditions may be expected to lie with a specified probability.

Reproducibility is one of the principles of the scientific method. It means that experiments must be described in sufficient detail to be accurately replicated by an independent researcher.In discussing measurement of dynamic parameters for process control, and specifically temperature, the phrase is often heard "just give me a number; it doesn't have to be accurate, only reproducible".?Unfortunately, this sets up the manufacturer for one of the most difficult faults to find and correct, intermittent poor quality. Here's how it happens:

Consider a process with two dynamic parameters, for example, a forging weld. For this example, the two parameters are pressure and temperature (a forging weld adds no material). Our sketch shows the surface of good quality, the boundary that separates good product from scrap. The arrowed lines show the natural variation of the parameters. (Even controlled parameters have some natural variation: besides the inaccuracies of the measuring instruments, the variables may not be measured everywhere in the product, or at all times, or the measurer/controller doesn't respond fast enough, etc.)

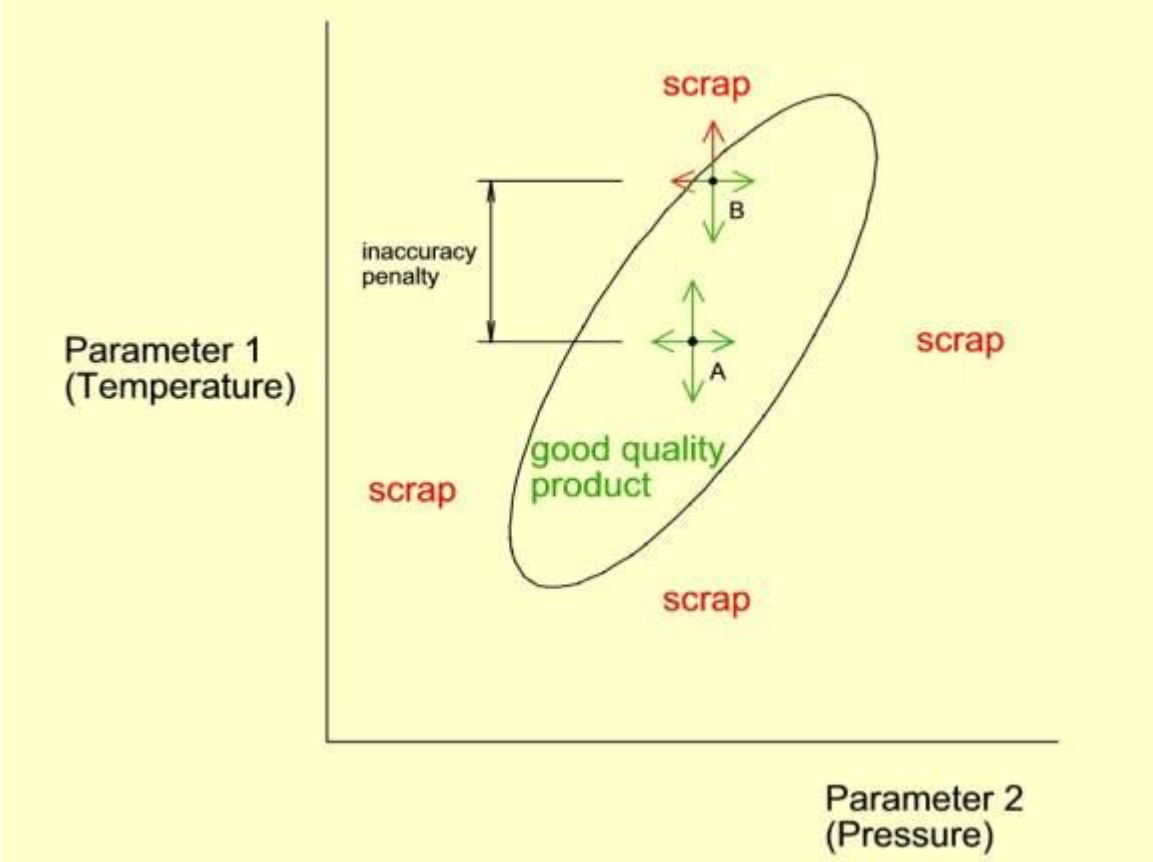


Figure: 1.19 Reproducibility

When the process is operating at Point A, all is well. However, at Point B there is intermittent scrap.?If the testing of the scrap product is sophisticated enough, it may determine that some failures are due to Parameter 2, as well as Parameter 1. This is a red herring, for we can see from the surface that the problem is entirely caused by a shift in the value of Parameter 1. (The small shift in Parameter 2 actually helps.)

How did the process get from point A to point B with the operators unaware of it? The manufacturer is relying on reproducibility, defined as: "the closeness of agreement among repeated measurements under the same operating conditions over time". Anything that changes the operating conditions destroys the reproducibility that the manufacturer is relying on. At Point B and unaware of it, the manufacturer must now pay the inaccuracy penalty.

Reproducibility is not realistically attainable in manufacturing. Change destroys it, and change is a constant. Raw materials, BTU content of fuel gases, phase angle of the electrical supply, mechanical/storage history of work pieces, process equipment wear or adjustment, operator methodology, ambient temperature and humidity, and more all change. Manufacturers periodically scramble to find out what is causing the problem, expending time, effort, raw product, and money, only to find the problem just goes away. However, it will be back. To ensure good quality, dynamic parameters should be measured as accurately as possible. Just as there is an inaccuracy penalty, there is an accuracy reward. Accurately mapping the surface of good quality allows the manufacturer to achieve it in different ways. Improved yield and quality are only the start of the benefits available; this knowledge can result in new products and processes, and the huge returns they promise.

1.7 RESOLUTION:

The smallest possible movement of a system. Also known as step size, resolution is determined by the feedback device and capabilities of the motion system. Theoretical resolution may exceed practical resolution. For example, in a ball-screw-based positioning system, a theoretical resolution of 4 nm can be obtained by combining a 4 mm/rev screw, 1000-line encoder, and an x1000 multiplier. The actual motion system will never be able to make a single 4 nm step due to friction, windup, and mechanical compliance. Therefore, the practical resolution is actually less. All of Aerotech's specifications are based on practical resolution. Refers to the sharpness and clarity of an image. The term is most often used to describe monitors, printers, and bit-mapped graphic images. In the case of dot-matrix and laser printers, the resolution indicates the number of dots per inch. For example, a 300-dpi (dots per inch) printer is one that is capable of printing 300 distinct dots in a line 1 inch long. This means it can print 90,000 dots per square inch.

In computers, resolution is the number of pixels (individual points of color) contained on a display monitor, expressed in terms of the number of pixels on the horizontal axis and the number on the vertical axis. The sharpness of the image on a display depends on the resolution and the size of the monitor. The same pixel resolution will be sharper on a smaller monitor and gradually lose sharpness on larger monitors because the same numbers of pixels are being spread out over a larger number of inches.

CHAPTER-2 DATA CLASSIFICATION

Data classification is the categorization of data for its most effective and efficient use. In a basic approach to storing computer data, data can be classified according to its critical value or how often it needs to be accessed, with the most critical or often-used data stored on the fastest media while other data can be stored on slower (and less expensive) media. This kind of classification tends to optimize the use of data storage for multiple purposes - technical, administrative, legal, and economic.

Data can be classified according to any criteria, not only relative importance or frequency of use. For example, data can be broken down according to its topical content, file type, operating platform, average file size in megabytes or gigabytes, when it was created, when it was last accessed or modified, which person or department last accessed or modified it, and which personnel or departments use it the most. A well-planned data classification system makes essential data easy to find. This can be of particular importance in risk management, legal discovery, and compliance with government regulations. Computer programs exist that can help with data classification, but in the end it is a subjective business and is often best done as a collaborative task that considers business, technical, and other points-of-view.

It is somewhat obvious that all of your data does not have the same importance to your business. Some of it is indeed mission critical; some of it is of limited or temporary value. One thing that is certainly true about data is that it is growing at very high, often exponential, rates. Your IT organization must have a process to align the value of data with the cost of storing and managing it. This process starts with a clear understanding of the business uses of data and provides a mechanism for storing it according to requirements established by business priorities. Optimizing the relationship between the cost of storing and managing data and the delivery of service levels for data access, recovery and discovery[1] is the objective of implementing an information lifecycle management (ILM) strategy. The foundation of an ILM implementation is the taxonomy established to classify data.

2.1 INFORMATION LIFECYCLE MANAGEMENT (ILM):

It is a sustainable storage strategy that balances the cost of storing and managing information with its changing business value. ILM provides a practical methodology for aligning storage costs with business priorities. Data classification is a process that defines the access, recovery and discovery characteristics of an enterprise's different sets of data, grouping them into logical categories to facilitate business objectives.

The collected data, also known as raw data or ungrouped data are always in an un organised form and need to be organized and presented in meaningful and readily comprehensible form in order to facilitate further statistical analysis. It is, therefore, essential for an investigator to condense a mass of data into more and more comprehensible and assimilable form. The process of grouping into different classes or sub classes according to some characteristics is known as classification, tabulation is concerned with the systematic arrangement and presentation of classified data. Thus classification is the first step in tabulation. For Example, letters in the post office are classified according to their destinations viz., Delhi, Madurai, Bangalore, Mumbai etc.,

2.2 WHY CLASSIFY DATA:

Data taxonomy serves multiple purposes. Effectively implemented, the taxonomy provides the cornerstone of an ILM strategy, supports linkage to other IT infrastructure issues (such as security) and can impart broader enterprise value as a basis of information asset management.

Information lifecycle management (ILM) is entirely dependent on effective data taxonomy. The taxonomy defines classes of data, each of which will be managed with a set of rules or policies. Without a solid, business-based taxonomy, the benefits of information lifecycle management cannot be achieved. The reason why data classification is necessary for ILM is relatively simple: the rules that define the information lifecycle management strategy must operate on classes of data to be economically possible. The alternative to classification is having as many rules as there are data objects. In an enterprise, this is a very unattractive proposition. Thus we seek to mass-customize our approach to managing data by grouping it into classes that have similar management requirements.

Organizations creating initial data taxonomy must carefully balance the number of data classes against the cost of maintaining and managing them. This generally can be handled in a manner similar to any mass-customization trade-off decision.

2.3 OBJECTS OF CLASSIFICATION:

The following are main objectives of classifying the data:

1. It condenses the mass of data in an easily assailable form.
2. It eliminates unnecessary details.
3. It facilitates comparison and highlights the significant aspect of data.
4. It enables one to get a mental picture of the information and helps in drawing inferences.
5. It helps in the statistical treatment of the information collected.
6. To define and describe digital image classification
7. To distinguish between supervised and unsupervised classification and describe how each is produced
8. To discuss the application of digital image classification data to remote sensing

2.4 TYPES OF CLASSIFICATION:

Statistical data are classified in respect of their characteristics. Broadly there are four basic types of classification namely

a) Chronological classification
b) Geographical classification
c) Qualitative classification
d) Quantitative classification

2.4.1 Chronological classification:

In chronological classification the collected data are arranged according to the order of time expressed in years, months, weeks, etc., The data is generally classified in ascending order of 39 time. For example, the data related with population, sales of a firm, imports and exports of a country are always subjected to chronological classification.

Example 5:

The estimates of birth rates in India during 1970 – 76 are

Year	1970	1971	1972	1973	1974	1975	1976
Birth Rate	36.8	36.9	36.6	34.6	34.5	35.2	34.2

2.4.2 Geographical classification:

In this type of classification the data are classified according to geographical region or place. For instance, the production of paddy in different states in India, production of wheat in different countries etc.,

Example 6:

Country	America China Denmark France India
Yield of wheat in (kg/acre)	1925 893 225 439 862

2.4.3 Qualitative classification:

In this type of classification data are classified on the basis of same attributes or quality like sex, literacy, religion, employment etc., such attributes cannot be measured along with a scale. For example, if the population to be classified in respect to one attribute, say sex, then we can classify them into two namely that of males and females. Similarly, they can also be classified into 'employed' or 'unemployed' on the basis of another attribute 'employment'. Thus when the classification is done with respect to one attribute, which is dichotomous in nature, two classes are formed, one possessing the attribute and the other not possessing the attribute. This type of classification is called simple or dichotomous classification. A simple classification may be shown as fewer than 40.

Population

(1)Male **(2) Female**

The classification, where two or more attributes are considered and several classes are formed, is called a manifold classification. For example, if we classify population simultaneously with respect to two attributes, e.g sex and employment, then population are first classified with respect to 'sex' into 'males' and 'females'. Each of these classes may then be further classified into ' employment' and ' unemployment' on the basis of attribute ' employment' and as such Population are classified into four classes namely.

(i) Male employed

(ii) Male unemployed

(iii) Female employed

(iv) Female unemployed

2.4.4 Quantitative classification:

Quantitative classification refers to the classification of data according to some characteristics that can be measured such as height, weight, etc., For example the students of a college may be Classified according to weight as given below.

Table 2.1 Quantitative Classification

Weight (in lbs)	No of Students
90-100	50
100-110	200
110-120	260
120-130	360
130-140	90
140-150	40
Total	1000

In this type of classification there are two elements, namely (i) the variable (i.e) the weight in the above example, and (ii) the frequency in the number of students in each class. There are 50 students having weights ranging from 90 to 100 lb, 200 students having weight ranging between 100 to 110 lb and so on.

All data and the information derived from data must be classified by the level of protection required. At a minimum data should be classified as either public or confidential. If additional classifications are needed to meet agency requirements, they should be defined clearly within the classification system. For example, some organizations have a category of "sensitive".

• Confidential: Protected by state or federal law.

• Sensitive: Not explicitly protected by law, but exposure could result in negative impact to government services, state government partners or citizens.

• Public: Data not included in a protected classification.

2.5 DATA HANDLING:

Data handling is defined as the process of ensuring that research data is stored, archived or disposed off in a safe and secure manner during and after the conclusion of a research project. Data collection anytime and everywhere has become the reality of our lives. After collection, data analysis is rather a more complex area requiring higher importance where in we have a novel work of data clustering. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A cluster is therefore a collection of objects, which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. The clustering problem has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. However, clustering is a difficult problem combinatorial, with lots of differences in assumptions and contexts of analysis. Data modeling puts clustering in a historical perspective rooted in mathematics, statistics, and numerical analysis. From a machine learning perspective clusters correspond to hidden patterns, the search for clusters is unsupervised learning, and the resulting system represents a data concept. From a practical perspective clustering plays an outstanding role in data mining applications such as scientific data exploration, information retrieval and text mining, spatial database applications, Web analysis, CRM, marketing, medical diagnostics, computational biology, control and many other applications. Data mining adds to clustering the complications of very large datasets with very many attributes of different types. This imposes unique computational requirements on relevant clustering algorithms. A variety of algorithms have recently emerged that meet these requirements and were successfully applied to real-life data mining problems. In recent years, the dramatic rise in the use of the web and the improvement in process industries in general have transformed our society into one that strongly depends on information. The huge amount of data that is generated by this process contains important information that accumulates daily in databases and is not easy to extract. Cluster analysis is the organization of a collection of patterns (usually represented as a vector of measurements, or a point in a multidimensional space) into clusters based on similarity.

Cluster analysis is thus a tool of discovery. It may reveal associations and structure in data, which, though not previously evident, nevertheless are sensible and useful once found. The

results of cluster analysis may contribute to the definition of a formal classification scheme, such as a taxonomy for related process variables, parameters or objects; or suggest statistical models with which to describe control; or indicate rules for assigning new cases to classes for control and analysis purposes; or provide measures of definition, variations and change in what previously were only broad concepts. The main requirements that a clustering algorithm should satisfy are the following:-

1 .Dealing with different types of attributes: There are various attributes of data that any clustering algorithm need to satisfy, the most general taxonomy being in common use distinguishes among numeric (continuous), ordinal, and nominal variables. A numeric variable can assume any value in R. An ordinal variable assumes a small number of discrete states, and these states can be compared.

2. Scalability to large datasets: The data sets could be in any possible range, varying between large extremes and they need to be normalized by the clustering algorithm.

3. Ability to work with high dimensional data: The data could be multidimensional varying from 1, 2……n.; depending on the application data on which clustering is being applied.

4. Ability to find clusters of irregular or arbitrary shape: The shape of clusters could be any arbitrary shapes. We prefer using Euclidean distance to get a circular shape of the clusters, but still shape of clusters can not be accurately defined

5. Handling outliers: The data points on the boundary of clusters need to be handled; this is done in hierarchical methods by associating the boundary points to one of the clusters. While in fuzzy clustering, we associate membership functions to the points lying on the boundary of clusters.

It can be shown that there is no absolute “best” criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. Typical pattern clustering activity involves the following steps:

(1) Pattern representation (optionally including feature extraction and/or selection),

(2) Definition of a pattern proximity measure appropriate to the data domain,

(3) Clustering or grouping,

(4) Data abstraction (if needed), and

(5) Assessment of output (if needed).

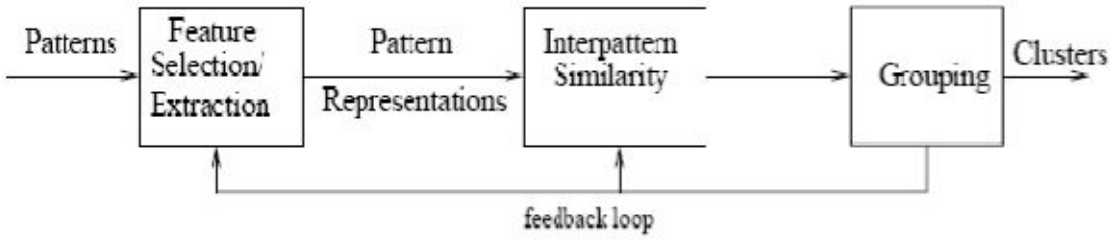


Figure 2.1: Stages of Clustering

Pattern representation refers to the number of classes, the number of available patterns, data, and the number, type, and scale of the features available to the clustering algorithm. Some of this information may not be controllable by the practitioner. Feature selection is the process of identifying the most effective subset of the original features to use in clustering. Feature extraction is the use of one or more transformations of the input features to produce new salient features. Either or both of these techniques can be used to obtain an appropriate set of features to use in clustering. Pattern proximity is usually measured by a distance function defined on pairs of patterns. A variety of distance measures are in use in the various communities. A simple distance measure like Euclidean distance can often be used to reflect dissimilarity between two patterns, whereas other similarity measures can be used to characterize the conceptual similarity between. The grouping step can be performed in a number of ways. The output clustering can be hard (a partition of the data into groups) or fuzzy (where each pattern has a variable degree of membership in each of the output clusters). Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partition clustering algorithms identify the partition that optimizes a clustering criterion. Additional techniques for the grouping operation include probabilistic and graph-theoretic clustering methods.

Data abstraction is the process of extracting a simple and compact representation of a data set. Here, simplicity is either from the perspective of automatic analysis (so that a machine can perform further processing efficiently) or it is human-oriented (so that the representation obtained is easy to comprehend and intuitively appealing). Cluster validity analysis, by contrast, is the assessment of a clustering procedure's output. Often this analysis uses a specific criterion of optimality; however, these criteria are usually arrived at subjectively. Hence, little in the way of 'gold standards' exist in clustering except in well-prescribed sub domains. Validity assessments are objective and are performed to determine whether the output is meaningful. A clustering structure is valid if it cannot reasonably have occurred by chance or as an artifact of a clustering algorithm. When statistical approaches to clustering are used, validation is accomplished by carefully applying statistical methods and testing hypotheses.

Quantitative features can be measured on a ratio scale (with a meaningful reference value, such as temperature), or on nominal or ordinal scales. One can also use structured features which are represented as trees, where the parent node represents a generalization of its child nodes. For example, a parent node "vehicle" may be a generalization of children labeled "cars," "buses," "trucks," and "motorcycles." Further, the node "cars" could be a generalization of cars of the type "Toyota," "Ford," "Benz," etc. Symbolic objects are defined by a logical conjunction of events. These events link values and features in which the features can take one or more values and all the objects need not be defined on the same set of features. It is often valuable to isolate only the most descriptive and discriminatory features in the input set, and utilize those features exclusively in subsequent analysis. Feature selection techniques identify a subset of the existing features for subsequent use, while feature extraction techniques compute new features from the original set. In either case, the goal is to improve classification performance and/or computational efficiency. Feature selection is a well-explored topic in statistical pattern recognition; however, in a clustering context (i.e., lacking class labels for patterns), the feature selection process is of necessity ad hoc, and might involve a trial-and-error process where various subsets of features are selected, the resulting patterns clustered, and the output evaluated using a validity index. In contrast, some of the popular feature extraction processes (e.g., principal components analysis do not depend on labeled data and can be used directly. Reduction of the number of features has an additional benefit, namely the ability to produce output that can be visually inspected by a human.

Clustering techniques can be applied to data that is quantitative (numerical), qualitative (categorical), or a mixture of both. In this thesis, the clustering of quantitative data is considered. The data are typically observations of some physical process. Each observation consists of *n* measured variables, grouped into an *n*-dimensional row vector.

$$x_k = [x_{k1}, x_{k2} \ldots\ldots\ldots\ldots x_{kn}]^T, x_k \in R^n$$

A set of *N* observations is denoted by $X = \{x_k \,|\, k = 1,2,\ldots\ldots,N\}$ and is represented as

an N X n matrix

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{Nn} \end{bmatrix}$$

In pattern recognition terminology, the rows of X are called patterns or objects, the columns are called the features or attributes, and X is called the pattern matrix. The meaning of the columns and rows of X with respect to reality depends on the context. In medical diagnosis, for instance, the rows of X may represent patients, and the columns are then symptoms, or laboratory measurements for the patients. When clustering is applied to the modeling and identification of dynamic systems, the rows of X contain samples of time signals, and the columns are, for instance, physical variables observed in the system. In order to represent the system's dynamics, past values of the variables are typically included in X as well. In system identification, the purpose of clustering is to find relationships between independent system variables, called the repressors, and future values of dependent variables, called the regress ands. One should, however, realize that the relations revealed by clustering are just causal associations among the data vectors, and as such do not yet constitute a prediction model of the given system.

2.6 DATA CLUSTERING:

Clustering can be considered the most important ***unsupervised learning*** problem; so, as every other problem of this kind, it deals with finding a ***structure*** in a collection of unlabeled data. A loose definition of clustering could be "the process of organizing objects into groups whose members are similar in some way". A ***cluster*** is therefore a collection of objects which are "similar" between them and are "dissimilar" to the objects belonging to other clusters. We can show this with a simple graphical example:

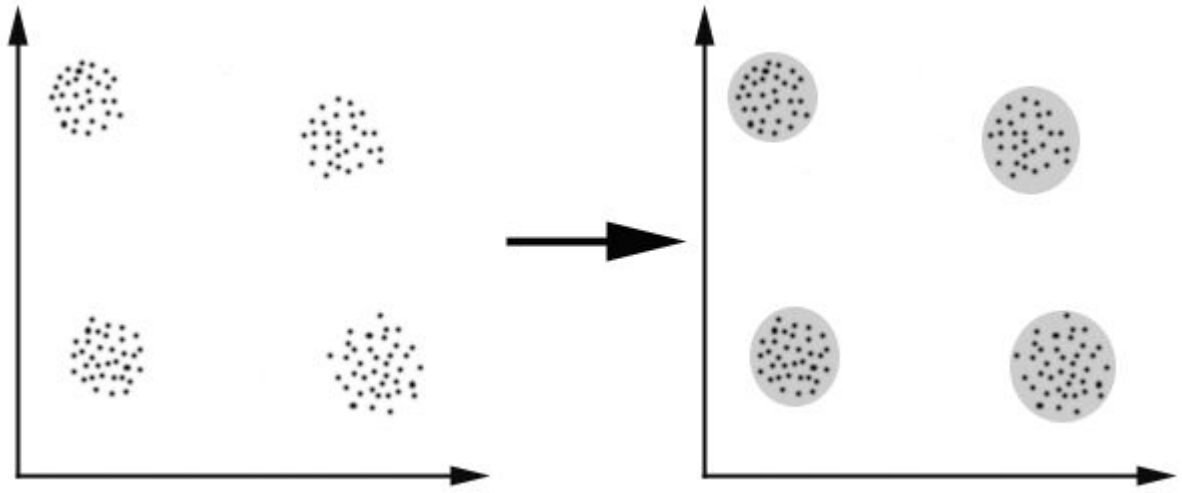


Figure: 2.2 Cluster Of Data

In this case we easily identify the 4 clusters into which the data can be divided; the similarity criterion is ***distance***: two or more objects belong to the same cluster if they are "close" according to a given distance (in this case geometrical distance). This is called ***distance-based clustering***. Another kind of clustering is ***conceptual clustering***: two or more objects belong to the same cluster if this one defines a concept ***common*** to all that objects. In other words, objects are grouped according to their fit to descriptive concepts, not according to simple similarity measures.

2.6.1 The Goals of Clustering:

So, the goal of clustering is to determine the intrinsic grouping in a set of unlabeled data. But how to decide what constitutes a good clustering? It can be shown that there is no absolute "best" criterion which would be independent of the final aim of the clustering. Consequently, it is the user which must supply this criterion, in such a way that the result of the clustering will suit their needs. For instance, we could be interested in finding representatives for homogeneous groups **(*data reduction*)**, in finding "natural clusters" and describe their unknown properties **(*"natural" data types*),** in finding useful and suitable groupings **(*"useful" data classes*)** or in finding unusual data objects **(*outlier detection*).**

2.6.2 Applications:

Clustering algorithms can be applied in many fields, for instance:

- ***Marketing***: finding groups of customers with similar behavior given a large database of customer data containing their properties and past buying records;
- ***Biology***: classification of plants and animals given their features;
- ***Libraries***: book ordering;
- ***Insurance***: identifying groups of motor insurance policy holders with a high average claim cost; identifying frauds;
- ***City-planning***: identifying groups of houses according to their house type, value and geographical location;
- ***Earthquake studies***: clustering observed earthquake epicenters to identify dangerous zones;
- ***WWW***: document classification; clustering weblog data to discover groups of similar access patterns.

2.6.3 Requirements:

The main requirements that a clustering algorithm should satisfy are:

- scalability;
- dealing with different types of attributes;
- discovering clusters with arbitrary shape;
- minimal requirements for domain knowledge to determine input parameters;
- ability to deal with noise and outliers;
- insensitivity to order of input records;
- high dimensionality;

- interpretability and usability.

2.6.4 Problems:

There are a number of problems with clustering. Among them:

- current clustering techniques do not address all the requirements adequately (and concurrently);
- dealing with large number of dimensions and large number of data items can be problematic because of time complexity;
- the effectiveness of the method depends on the definition of "distance" (for distance-based clustering);
- if an ***obvious*** distance measure doesn't exist we must "define" it, which is not always easy, especially in multi-dimensional spaces;
- The result of the clustering algorithm (that in many cases can be arbitrary itself) can be interpreted in different ways.

2.6.5 Clustering Algorithms:

Clustering algorithms may be classified as listed below:

- Exclusive Clustering
- Overlapping Clustering
- Hierarchical Clustering
- Probabilistic Clustering

In the first case data are grouped in an exclusive way, so that if a certain datum belongs to a definite cluster then it could not be included in another cluster. A simple example of that is shown in the figure below, where the separation of points is achieved by a straight line on a bi-dimensional plane. On the contrary the second type, the overlapping clustering, uses fuzzy sets to cluster data, so that each point may belong to two or more clusters with different degrees of membership. In this case, data will be associated to an appropriate membership value.

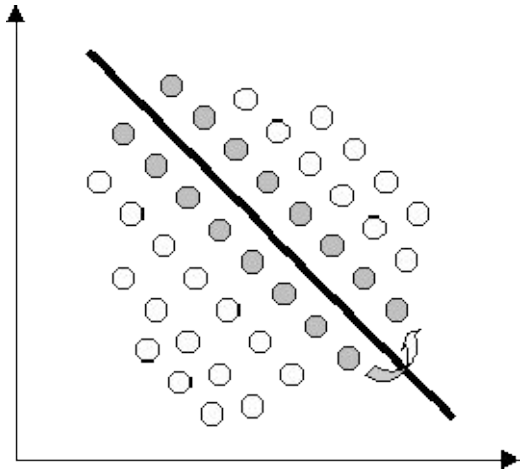


Figure: 2.3 Exclusive Cluster

Instead, a hierarchical clustering algorithm is based on the union between the two nearest clusters. The beginning condition is realized by setting every datum as a cluster. After a few iterations it reaches the final clusters wanted. Finally, the last kind of clustering uses a completely probabilistic approach.

2.6.6 K-Means Clustering:

K-means (Mac Queen, 1967) is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters) fixed a priori. The main idea is to define k centroids, one for each cluster. These centroids should be placed in a cunning way because of different location causes different result. So, the better choice is to place them as much as possible far away from each other. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bar centers of the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done between the same data set points and the nearest new centroid. A loop has been generated. As a result of this loop we may notice that the k centroids change their location step by step until no more changes are done. In other words centroids do not move any more. Finally, this algorithm aims at minimizing an ***objective function***, in this case a squared error function.

The objective function

$$J = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

where $\left\| x_i^{(j)} - c_j \right\|^2$ is a chosen distance measure between a data point $x_i^{(j)}$ and the cluster centre c_j, is an indicator of the distance of the ***n*** data points from their respective cluster centers.

The algorithm is composed of the following steps:

1. ***Place K points into the space represented by the objects that are being clustered. These points represent initial group centroids.***
2. ***Assign each object to the group that has the closest centroid.***
3. ***When all objects have been assigned, recalculate the positions of the K centroids.***
4. ***Repeat Steps 2 and 3 until the centroids no longer move. This produces a separation of the objects into groups from which the metric to be minimized can be calculated.***

Although it can be proved that the procedure will always terminate, the k-means algorithm does not necessarily find the most optimal configuration, corresponding to the global objective function minimum. The algorithm is also significantly sensitive to the initial randomly selected cluster canters. The k-means algorithm can be run multiple times to reduce this effect. K-means is a simple algorithm that has been adapted to many problem domains. As we are going to see, it is a good candidate for extension to work with fuzzy feature vectors.

Remarks:

this is a simple version of the k-means procedure. It can be viewed as a greedy algorithm for partitioning the n samples into k clusters so as to minimize the sum of the squared distances to the cluster centers. It does have some weaknesses:

- The way to initialize the means was not specified. One popular way to start is to randomly choose k of the samples.
- The results produced depend on the initial values for the means, and it frequently happens that suboptimal partitions are found. The standard solution is to try a number of different starting points.

- It can happen that the set of samples closest to $\mathbf{m}_i$ is empty, so that $\mathbf{m}_i$ cannot be updated. This is an annoyance that must be handled in an implementation, but that we shall ignore.
- The results depend on the metric used to measure $\| \mathbf{x} - \mathbf{m}_i \|$. A popular solution is to normalize each variable by its standard deviation, though this is not always desirable.
- The results depend on the value of k.

This last problem is particularly troublesome, since we often have no way of knowing how many clusters exist. In the example shown above, the same algorithm applied to the same data produces the following 3-means clustering. Is it better or worse than the 2-means clustering?

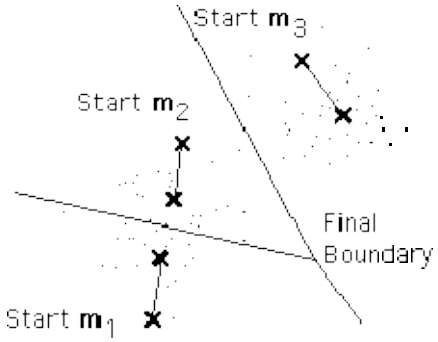


Figure: 2.4 K Mean Clustering

Unfortunately there is no general theoretical solution to find the optimal number of clusters for any given data set. A simple approach is to compare the results of multiple runs with different k classes and choose the best one according to a given criterion (for instance the Schwarz Criterion - see Moore's slides), but we need to be careful because increasing k results in smaller error function values by definition, but also an increasing risk of over fitting.

2.6.7 Fuzzy C-Means Clustering:

Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method (developed by Dunn in 1973 and improved by Bezdek in 1981) is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - c_j \right\|^2 , \qquad 1 \leq m < \infty$$

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 ith of d-dimensional measured data, c_j is the d-dimension center of the cluster, and ||*|| is any norm expressing the similarity between any measured data and the center. 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:

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

This iteration will stop when $\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{(k)} \right| \right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, whereas k is the iteration steps. This procedure converges to a local minimum or a saddle point of J_m. The algorithm is composed of the following steps:

1. ***Initialize U=[u_{ij}] matrix, $U^{(0)}$***
2. ***At k-step: calculate the centers vectors $C^{(k)}$=[c_j] with $U^{(k)}$***

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m \cdot x_i}{\sum_{i=1}^{N} u_{ij}^m}$$

3. ***Update $U^{(k)}$, $U^{(k+1)}$***

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}$$

4. ***If || $U^{(k+1)}$ - $U^{(k)}$||< ε then STOP; otherwise return to step 2.***

Remarks: As already told, data are bound to each cluster by means of a Membership Function, which represents the fuzzy behavior of this algorithm. To do that, we simply have to build an appropriate matrix named U whose factors are numbers between 0 and 1, and represent the

degree of membership between data and centers of clusters.

For a better understanding, we may consider this simple mono-dimensional example. Given a certain data set, suppose to represent it as distributed on an axis. The figure below shows this:

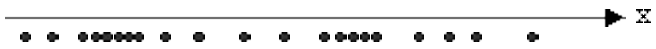


Figure: 2.5 Mono Dimentional Data

In the first approach shown in this tutorial - the k-means algorithm - we associated each datum to a specific centroid; therefore, this membership function looked like this:

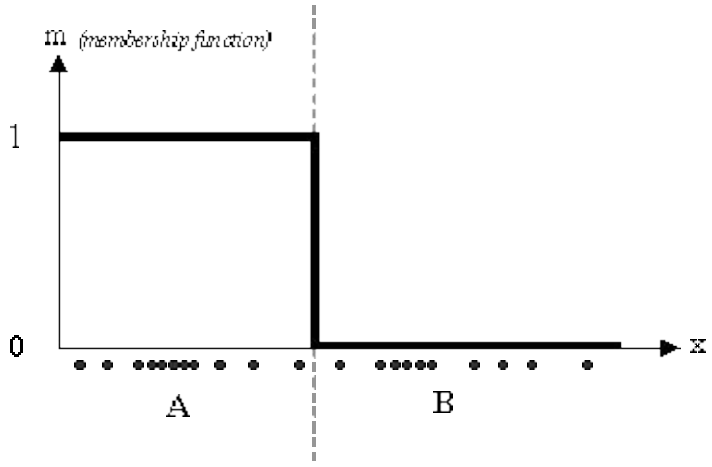


Figure: 2.6 Membership Function

In the FCM approach, instead, the same given datum does not belong exclusively to a well defined cluster, but it can be placed in a middle way. In this case, the membership function follows a smoother line to indicate that every datum may belong to several clusters with different values of the membership coefficient.

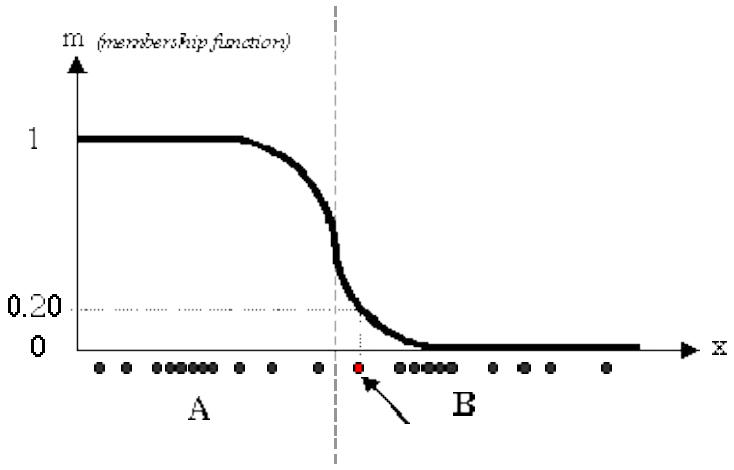


Figure: 2.7 Membership Values

In the figure above, the datum shown as a red marked spot belongs more to the B cluster rather than the A cluster. The value 0.2 of 'm' indicates the degree of membership to A for such datum. Now, instead of using a graphical representation, we introduce a matrix U whose factors are the ones taken from the membership functions:

$$U_{MC} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ .. & .. \\ 0 & 1 \end{bmatrix} \qquad U_{MC} = \begin{bmatrix} 0.8 & 0.2 \\ 0.3 & 0.7 \\ 0.6 & 0.4 \\ .. & .. \\ 0.9 & 0.1 \end{bmatrix}$$

(a) (b)

The number of rows and columns depends on how many data and clusters we are considering. More exactly we have C = 2 columns (C = 2 clusters) and N rows, where C is the total number of clusters and N is the total number of data. The generic element is so indicated: $\boldsymbol{u_{ij}}$.

In the examples above we have considered the k-means (a) and FCM (b) cases. We can notice that in the first case (a) the coefficients are always unitary. It is so to indicate the fact that each datum can belong only to one cluster. Other properties are shown below:

- $u_{ij} \in [0,1] \qquad \forall i, j$
- $\sum_{j=1}^{C} u_{ik} = 1 \qquad \forall i$

- $0 < \sum_{i=1}^{N} u_{ij} < N \quad \forall N$

2.7 DATA CLASSIFICATION BENEFITS:

In a broad sense, data classification enables information lifecycle management and information asset management. One could easily argue that neither is economically possible without the solid underpinning of high-quality data classification taxonomy, coupled with solid data classification project implementation and ongoing change management. Data classification is the enabler of a mass-customized approach to implementing information lifecycle management or storage optimization rules. As such, data classification drives organizational agility by facilitating a set of processes that are regularly reviewed and refined. It enables a categorized approach to managing to the requirements of disparate business processes, and business processes that share business data objects. Further, from an information lifecycle management perspective, data classification can reduce business risks by providing that the appropriate data is managed with the appropriate standards for compliance, retention, protection and security.

Another key element of the data classification process is defining your various classification levels. There are no firm rules about the titles and types of classifications. However, the classifications should be clear enough so it is easy to decide how to classify the data once the process is underway. Many organizations use a classic military model, such as "confidential," "secret" and "top secret." Others are adding classifications specifically for privacy data.

Some companies define data by business process, restricting access only to those who participate in the business process. Think of an "eyes only" style of classification. For example, a research division may restrict information to one group within the division, classifying it as the New Molecule Group classification. While this creates a greater volume of classifications, it is more business process-specific. There is technology available today that allows for the automatic classification of information based on a business process-style classification system, as well as a military-style system.

CHAPTER 3 ARTIFICIAL INTELLIGENCE AND FUZZY LOGIC

"Fuzzy logic is basically a multivalued logic that allows intermediate values to be defined between conventional evaluations like yes/no, true/false, black/white, etc. Notions like rather warm or pretty cold can be formulated mathematically and processed by computers."

Fuzzy logic was first invented as a representation scheme and calculus for uncertain or vague notions. It allows more human-like interpretation and reasoning in machines by resolving intermediate categories between notations such as true/false, hot/cold etc used in Boolean logic In this context, Fuzzy logic is a problem-solving control system methodology that lends itself to implementation in systems ranging from simple, small, embedded micro-controllers to large, networked, multi-channel PC or workstation-based data acquisition and control systems. It can be implemented in hardware, software, or a combination of both. FL provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information. FL's approach to control problems mimics how a person would make decisions, only much faster.

Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. Fuzzy provides a remarkably simple way to draw definite conclusions from vague, ambiguous or imprecise information. In a sense, fuzzy logic resembles human decision making with its ability to work from approximate data and find precise solutions. Unlike classical logic, which requires a deep understanding of a system, exact equations, and precise numeric values, Fuzzy logic incorporates an alternative way of thinking, which allows modeling complex systems using a higher level of abstraction originating from our knowledge and experience. Fuzzy logic allows expressing this knowledge with subjective concepts such as very hot, bright red, and a long time, which are mapped into exact numeric ranges.

Fuzzy logic is a paradigm for an alternative design methodology, which can be applied in developing both linear and non-linear systems for embedded control. Fuzzy logic provides an alternative solution to non-linear control because it is closer to the real world. Rules, membership

functions, and the inference process which results in improved performance, simpler implementation, and reduced design costs handle non-linearity.

By using fuzzy logic, designers can realize lower development costs, superior features, and better end product performance. Furthermore, products can be brought to market faster and more cost-effectively. Fuzzy logic has been gaining increasing acceptance during the past few years. There are over two thousand commercially available products using Fuzzy logic, ranging from washing machines to high-current trains. Nearly every application can potentially realize some of the benefits of Fuzzy logic, such as performance, simplicity, lower cost, and productivity.

At present Fuzzy logic are used in many applications. Today it's being used in a simple washing machine that we use in our houses to automatic focusing system of an industry.

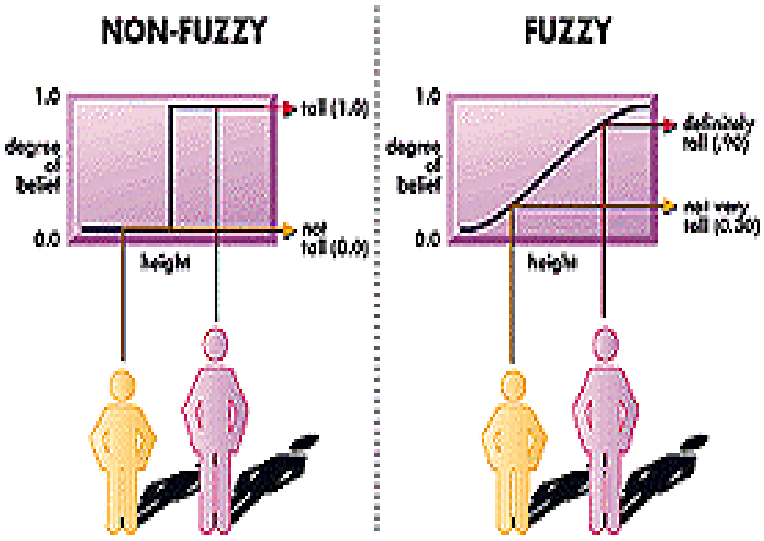
The few applications of fuzzy logic are:

- Washing Machine,
- Automatic Focusing System,
- Servo Motor Force Control,
- Dead Time Compensator (glass),
- Error Compensator (glass), Reactor Temperature Control,
- Two Stage Inverted Pendulum,
- Automated Manufacturing,
- Camera Auto focus,
- Servo Motor Force Control,
- Glass Melting Furnace Control,
- Air Conditioner Control,
- Reactor Control,
- CAR Automatic Transmissions

The concept of Fuzzy logic (FL) was conceived by Lotfi Zadeh, a professor at the University of California at Berkley, and presented it not as a control methodology, but as a way of processing data by allowing partial set membership rather than crisp set membership or non-membership. This approach to set theory was not applied to control systems until the 70's due to insufficient small-computer capability prior to that time. Professor Zadeh reasoned that people do not require

precise, numerical information input, and yet they are capable of highly adaptive control. If feedback controllers could be programmed to accept noisy, imprecise input, they would be much more effective and perhaps easier to implement. Unfortunately, U.S. manufacturers have not been so quick to embrace this technology while the Europeans and Japanese have been aggressively building real products around Professor Zadeh's paper on fuzzy sets introduced the concept of a class with un sharp boundaries and marked the beginning of a new direction by providing a basis for a qualitative approach to the analysis of complex systems in which linguistic rather than numerical variables are employed to describe system behavior and performance. This approach canters on building better models of human reasoning and decision-making. The ordinary fuzzy logic models are termed as Type I Fuzzy logic models.

3.1 FUZZY LOGIC Vs CONVENTIONAL CONTROL SYSTEMS



:

Figure: 3.1 Comparission Non-Fuzzy System And Fuzzy System

Fuzzy logic incorporates a simple, rule-based if X and Y then Z approach to a solving control problem rather than attempting to model a system mathematically. The Fuzzy logic model is empirically based, relying on an operator's experience rather than their technical understanding of the system. For example, rather than dealing with temperature control in terms such as "SP

=500F", "T <1000F", or "210C <TEMP <220C", terms like "IF (process is too cool) AND (process is getting colder) THEN (add heat to the process)" or "IF (process is too hot) AND (process is heating rapidly) THEN (cool the process quickly)" are used. These terms are imprecise and yet very descriptive of what must actually happen. Consider what you do in the shower if the temperature is too cold: you will make the water comfortable very quickly with little trouble. FL is capable of mimicking this type of behavior but at very high rate.

Additional benefits of fuzzy logic include its simplicity and its flexibility. Fuzzy logic can handle problems with imprecise and incomplete data, and it can model nonlinear functions of arbitrary complexity. "If you don't have a good plant model, or if the system is changing, then fuzzy will produce a better solution than conventional control techniques". A fuzzy system can create to match any set of input-output data. The Fuzzy logic Toolbox makes this particularly easy by supplying adaptive techniques such as adaptive neuro-fuzzy inference systems (ANFIS) and fuzzy subtractive clustering. Fuzzy logic models, called fuzzy inference systems, consist of a number of conditional "if-then" rules. For the designer who understands the system, these rules are easy to write, and as many rules as necessary can be supplied to describe the system adequately (although typically only a moderate number of rules are needed).

Not only do the rule-based approach and flexible membership function scheme make fuzzy systems straightforward to create, but they also simplify the design of systems and ensure that you can easily update and maintain the system over time.

3.2 WHY USE FL?

FL offers several unique features that make it a particularly good choice for many control problems.

1) It is inherently robust since it does not require precise, noise-free inputs and can be programmed to fail safely if a feedback sensor quits or is destroyed. The output control is a smooth control function despite a wide range of input variations.

2) Since the FL controller processes user-defined rules governing the target control system, it can be modified and tweaked easily to improve or drastically alter system performance. New sensors can easily be incorporated into the system simply by generating appropriate governing rules.

3) FL is not limited to a few feedback inputs and one or two control outputs, nor is it necessary to measure or compute rate-of-change parameters in order for it to be implemented. Any sensor data that provides some indication of a system's actions and reactions is sufficient. This allows the sensors to be inexpensive and imprecise thus keeping the overall system cost and complexity low.

4) Because of the rule-based operation, any reasonable number of inputs can be processed and numerous outputs generated, although defining the rule base quickly becomes complex if too many inputs and outputs are chosen for a single implementation since rules defining their interrelations must also be defined. It would be better to break the control system into smaller chunks and use several smaller FL controllers distributed on the system, each with more limited responsibilities.

5) FL can control nonlinear systems that would be difficult or impossible to model mathematically.

3.3 HOW IS FL USED?

1) Define the control objectives and criteria: What am I trying to control? What do I have to do to control the system? What kind of response do I need? What are the possible (probable) system failure modes?

2) Determine the input and output relationships and choose a minimum number of variables for input to the FL engine (typically error and rate-of-change-of-error).

3) Using the rule-based structure of FL, break the control problem down into a series of IF X AND Y THEN Z rules that define the desired system output response for given system input conditions. The number and complexity of rules depends on the number of input parameters that

are to be processed and the number fuzzy variables associated with each parameter. If possible, use at least one variable and its time derivative. Although it is possible to use a single, instantaneous error parameter without knowing its rate of change, this cripples the system's ability to minimize overshoot for a step inputs.

4) Create FL membership functions that define the meaning (values) of Input/Output terms used in the rules.

5) Test the system, evaluate the results, tune the rules and membership functions, and retest until satisfactory results are obtained.

3.4 FUZZY OPERATION:

FL requires some numerical parameters in order to operate such as what is considered significant error and significant rate-of-change-of-error, but exact values of these numbers are usually not critical unless very responsive performance is required in which case empirical tuning would determine them.

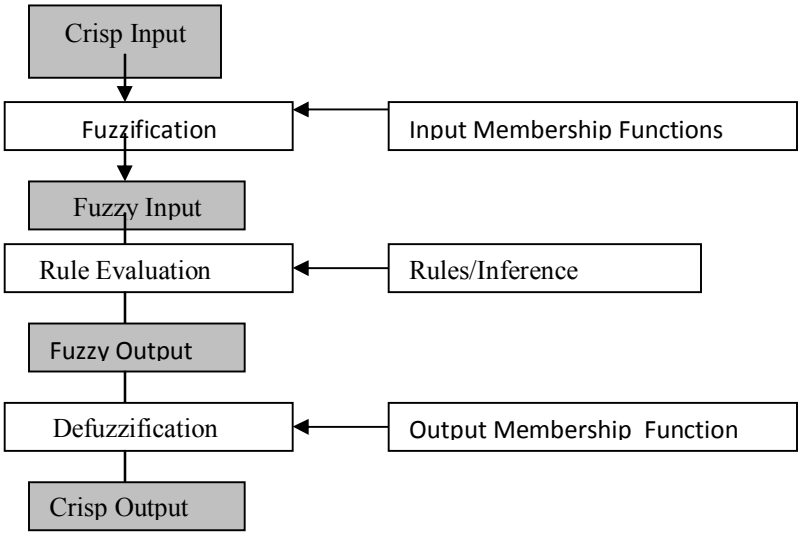


Figure 3.2: Fuzzy Operation

For example, a simple temperature control system could use a single temperature feedback sensor whose data is subtracted from the command signal to compute "error" and then time-differentiated to yield the error slope or rate-of-change-of-error, hereafter called "error-dot". Error might have units of degs F and a small error considered to be 2F while a large error is 5F. The "error-dot" might then have units of degs/min with a small error-dot being 5F/min and a large one being 15F/min. These values don't have to be symmetrical and can be "tweaked" once the system is operating in order to optimize performance. Generally, FL is so forgiving that the system will probably work the first time without any tweaking.

3.5 HOW DOES FUZZY LOGIC WORK?

In order to understand this let us see a simplified diagram of a thermostat controlling a heater fan.

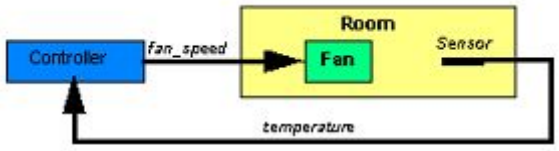


Figure: 3.3 Block Diagram of Fuzzy Logic

The room temperature detected through a sensor is input to a controller, which outputs a control force to adjust the heater fan current. A conventional thermostat works like an on-off switch. If we set it at 78oF then the heater is activated only when the temperature falls below 75oF. When it reaches 81oF the heater is turned off. As a result the desired room temperature is either too warm or too hot.

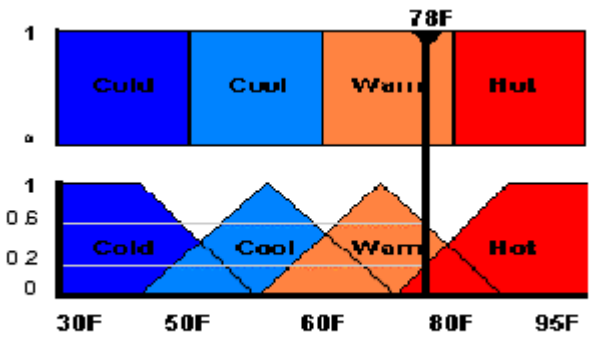


Figure: 3.4 Memebership Assignment

A fuzzy thermostat works where the temperature is treated as a series of overlapping ranges. For example, 78oF is 60% warm and 20% hot. The controller is programmed with simple if-then rules that tell the heater fan how fast to run. As a result, when the temperature changes the fan current will continuously adjust to keep the temperature at the desired level.

Our first step in designing such a fuzzy controller is to characterize the range of values for the input and output variables of the controller. Then we assign labels such as cool for the temperature and high for the fan current, and we write a set of simple English-like rules to control the system. Inside the controller all temperature regulating actions will be based on how the current room temperature falls into these ranges and the rules describing the system behavior. The controller's output will vary continuously to adjust the fan current. The temperature controller described above can be defined in four simple rules:

IF temperature IS cold THEN fan current IS high

IF temperature IS cool THEN fan current IS medium

IF temperature IS warm THEN fan current IS low

IF temperature IS hot THEN fan current IS zero

Here the linguistic variables cool; warm, high, etc. are labels. These triangular shaped values are called membership functions. A fuzzy controller works similar to a conventional system: it accepts an input value, performs some calculations, and generates an output value. This process is called the Fuzzy Inference Process and works in three steps:

a) FUZZIFICATION: where a crisp input is translated into a fuzzy value.

b) RULE EVALUATION: where the fuzzy output truth values are computed, and

c) DEFUZZIFICATION: where the fuzzy output is translated to a crisp value.

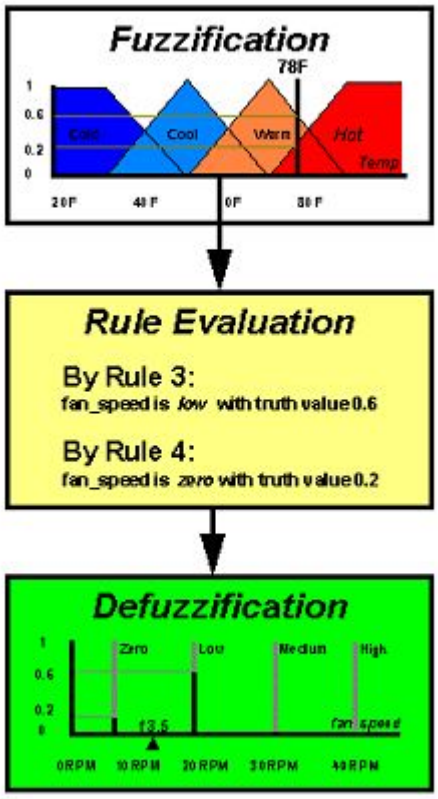


Figure: 3.5 Fuzzy Inference Process

During the Fuzzification step the crisp temperature value of 78oF is input and translated into fuzzy truth values. For this example, 78oF is fuzzified into warm with truth value 0.6 (or 60%) and hot with truth value 0.2 (or 20%). During the rule evaluation step the entire set of rules is evaluated and some rules may fire up. For 78oF only the last two of the four rules will fire. Specifically, using rule three the fan current will be low with degree of truth 0.6.

Similarly, using rule four the fan current will be zero with degree of truth 0.2. During the defuzzification step the 60% low and 20% zero labels are combined using a calculation method called the Center of Gravity (COG) in order to produce the crisp output value of 13.5 RPM for the fan current.

3.6 FUZZY SETS:

The very basic notion of fuzzy systems is a fuzzy (sub) set. In classical mathematics we are familiar with what we call crisp sets. Fuzzy Set Theory can describe this natural phenomenon more accurately. Fig. below shows how fuzzy sets quantifying the same information can describe this natural drift.

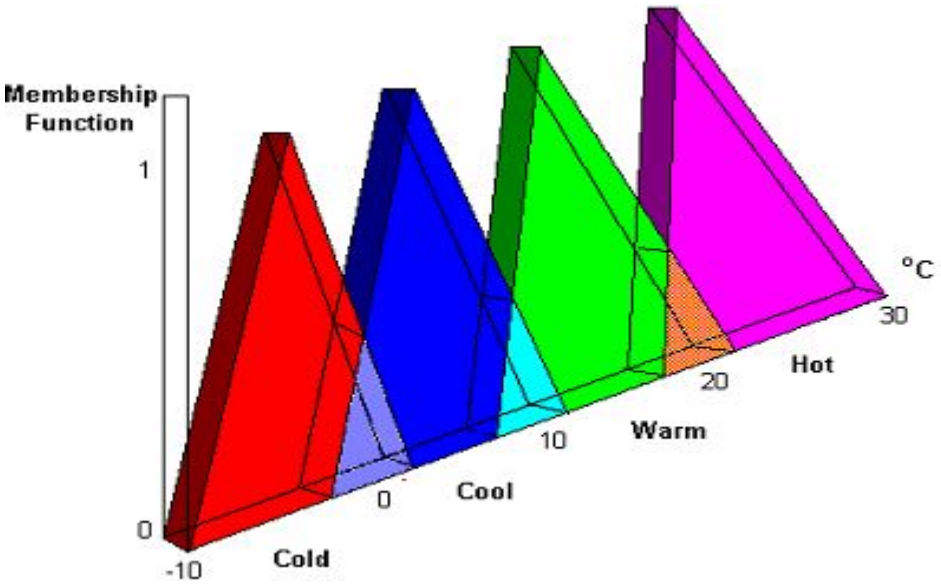


Figure: 3.6 Fuzzy Set

For example, the possible interferometer coherence values are the set *X* of all real numbers between 0 and 1. From this set *X* a subset *A* can be defined, (e.g. all values $0 < g < 0.2$). The characteristic function of A, (i.e. this function assigns a number 1 or 0 to each element in *X,* depending on whether the element is in the subset *A* or not) is shown in Fig.

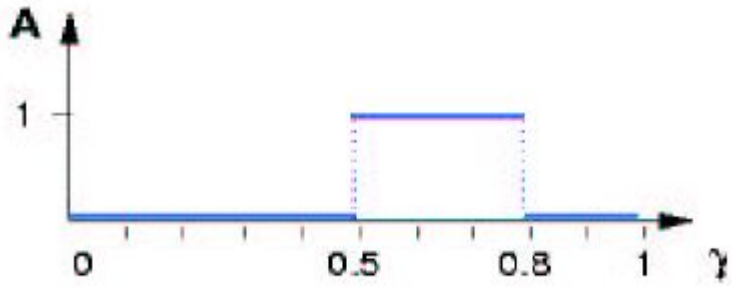


Figure: 3.7 Membership Value

The elements which have been assigned the number 1 can be interpreted as the elements that are in the set *A* and the elements which have assigned the number 0 as the elements that are not in the set *A*. This concept is sufficient for many areas of applications, but it can easily be seen, that it lacks in flexibility for some applications like classification of remotely sensed data analysis.

For example it is well known that water shows low interferometer coherence g in images. Since g starts at 0, the lower range of this set ought to be clear. The upper range, on the other hand, is rather hard to define. As a first attempt, we set the upper range to 0.2. Therefore we get B as a crisp interval B= [0, 0.2]. But this means that a g value of 0.20 is low but a g value of 0.21 not.

Obviously, this is a structural problem, for if we moved the upper boundary of the range from g =0.20 to an arbitrary point we can pose the same question. A more natural way to construct the set *B* would be to relax the strict separation between *low* and *not low*. This can be done by allowing not only the (crisp) decision Yes/No, but more flexible rules like “fairly low".

A fuzzy set allows us to define such a notion. The aim is to use fuzzy sets in order to make computers more intelligent; therefore, the idea above has to be coded more formally. In the example, all the elements were coded with 0 or 1. A straight way to generalize this concept is to allow more values between 0 and 1. In fact infinitely many alternatives can be allowed between 0 and 1, namely the unit interval I = [0, 1]. The interpretation of the numbers, now assigned to all elements is much more difficult. Of course, again the number 1 assigned to an element means that the element is in the set *B* and 0 means that the element is definitely not in the set *B*. All other values mean a gradual membership to the set *B*. This is shown in Fig. 2. The membership function is a graphical representation of the magnitude of participation of each input. It associates a weighting with each of the inputs that are processed, define functional overlap between inputs, and ultimately determines an output response. The rules use the input membership values as weighting factors to determine their influence on the fuzzy output sets of the final output conclusion.

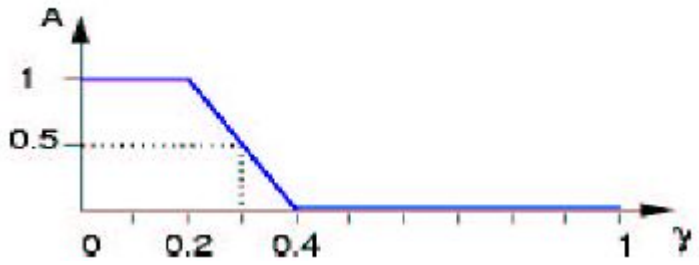


Figure: 3.8 Characteristic Function of fuzzy set

The membership function, operating in this case on the fuzzy set of interferometer coherence g, returns a value between 0.0 and 1.0. For example, an interferometer coherence g of 0.3 has a membership of 0.5 to the set *low coherence* (see Fig. 2).It is important to point out the distinction between fuzzy logic and probability. Both operate over the same numeric range, and have similar

Values: 0.0 representing False (or non_ membership), and 1.0 representing *True* (or full_ membership). However, there is a distinction to be made between the two statements: The probabilistic approach yields the natural language statement, "There is an 50% chance that g is low," while the fuzzy terminology corresponds to "g's degree of membership within the set of low interferometer coherence is 0.50." The semantic difference is significant: the first view supposes that g is or is not low; it is just that we only have a 50% chance of knowing which set it is in. By contrast, fuzzy terminology supposes that g is "more or less" low, or in some other term corresponding to the value of 0.50.

3.7 WHY USE FUZZY LOGIC:

- An Alternative Design Methodology Which Is Simpler, And Faster
    - Fuzzy logic reduces the design development cycle
    - Fuzzy logic simplifies design complexity
    - Fuzzy logic improves time to market

- A Better Alternative Solution To Non-Linear Control
    - Fuzzy logic improves control performance
    - Fuzzy logic simplifies implementation
    - Fuzzy logic reduces hardware costs

Fuzzy logic is a paradigm for an alternative design methodology, which can be applied in developing both linear and non-linear systems for embedded control. By using fuzzy logic, designers can realize lower development costs, superior features, and better end product performance. Furthermore, products can be brought to market faster and more cost-effectively.

3.8 FUZZY LOGIC SIMPLIFIES IMPLEMENTATION:

The one input temperature controller presented so far has helped us illustrate some fundamental concepts, however real life control is much more complex in nature. Most control applications have multiple inputs and require modeling and tuning of a large number of parameters, which makes implementation very tedious and time consuming. Fuzzy rules can help you simplify implementation by combining multiple inputs into single if-then statements while still handling non-linearity

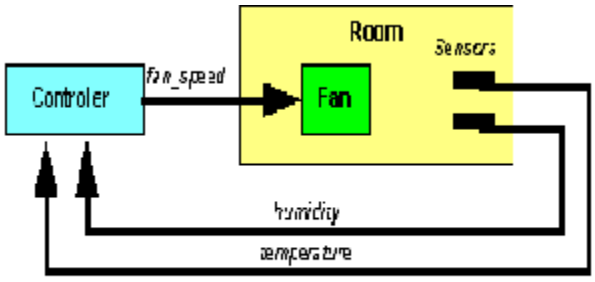


Figure:3.9 Implementation Using Fuzzy Logic

Consider a modified version of the temperature controller example, with two inputs, temperature and humidity and the same output, fan current.

IF temperature IS cold AND humidity IS high THEN fan_spd IS high
IF temperature IS cool AND humidity IS high THEN fan_spd IS medium

IF temperature IS warm AND humidity IS high THEN fan_spd IS low
IF temperature IS hot AND humidity IS high THEN fan_spd IS zero

IF temperature IS cold AND humidity IS med THEN fan_spd IS medium
IF temperature IS cool AND humidity IS med THEN fan_spd IS low
IF temperature IS war m AND humidity IS med THEN fan_spd IS zero
IF temperature IS hot AND humidity IS med THEN fan_spd IS zero

IF temperature IS cold AND humidity IS low THEN fan_spd IS medium
IF temperature IS cool AND humidity IS low THEN fan_spd IS low
IF temperature IS warm AND humidity IS low THEN fan_spd IS zero

A linear approximation requires handling each input separately, which multiplies design effort. Similarly, a piecewise linear approach requires the design of several controllers and is costly to implement. A lookup table seems more appropriate for this problem but it takes time to develop, debug and tune. For example, if we assume that each input requires eight bits, a lookup table would require 64K entries which make it very time consuming to implement.

Another example of simplicity is the classical control problem of the double stage inverted pendulum. Using conventional programming, this problem is extremely difficult, or impossible to implement. Aptronix has demonstrated a physical model of the 2-stage inverted pendulum, which was accomplished using only 30 rules. The software portion of the project took only two days to develop.

3.9 FUZZIFICATION:

Through the use of membership functions defined for each fuzzy set for each linguistic variable, the degree of membership of a crisp value in each fuzzy set is determined. As an example below, the numerical variable *age*, which has a given value of 25.0, was fuzzified using the triangular membership functions defined for each fuzzy set for linguistic variable *age*. As a result of fuzzification, linguistic variable *age* has linguistic values of "young" with a degree of membership of 0.666, "quite old" with a degree of 0.333, and for the remaining linguistic values with a degree of 0.0.

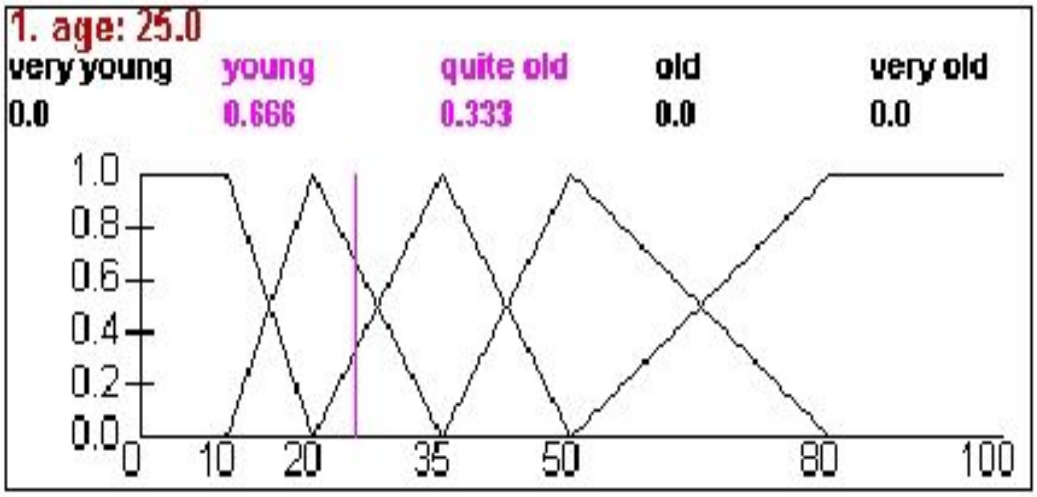


Figure: 3.10 Sample Fuzzification of Crisp Inputs

In a fuzzy expert system application, each input variable's crisp value is first fuzzified into linguistic values before the inference engine proceeds in processing with the rule base.

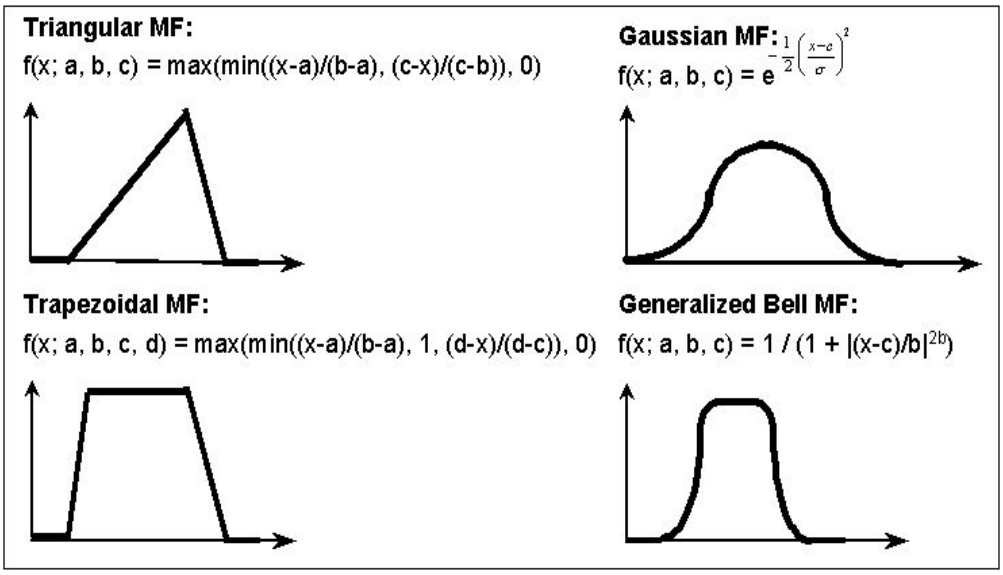


Figure: 3.11 Functions Using For Fuzzification

There are many types of membership functions. Some of the more common ones are triangular MFs (such as the functions in the figure above), trapezoidal MFs, Gaussian MFs, and generalized bell MFs.

3.10 APPLICATIONS:

Areas in which fuzzy logic has been successfully applied are often quite concrete. The first major commercial application was in the area of cement kiln control, an operation which requires that an operator monitor four internal states of the kiln, control four sets of operations, and dynamically manage 40 or 50 "rules of thumb" about their interrelationships, all with the goal of controlling a highly complex set of chemical interactions. One such rule is "If the oxygen percentage is rather high and the free-lime and kiln-drive torque rate is normal, decrease the flow of gas and slightly reduce the fuel rate" A complete accounting of this very successful system can be found in Umbers and King. The objection has been raised that utilizing fuzzy systems in a dynamic control environment raises the likelihood of encountering difficult stability problems: since in control conditions the use of fuzzy systems can roughly correspond to using thresholds, there must be significant care taken to insure that oscillations do not develop in the "dead spaces" between threshold triggers. This seems to be an important area for future research.

Other applications which have benefited through the use of fuzzy systems theory have been information retrieval systems, a navigation system for automatic cars, a predicative fuzzy-logic controller for automatic operation of trains, laboratory water level controllers, controllers for robot arc-welders, feature-definition controllers for robot vision, graphics controllers for automated police sketchers, and more. Expert systems have been the most obvious recipients of the benefits of fuzzy logic, since their domain is often inherently fuzzy. Examples of expert systems with fuzzy logic central to their control are decision-support systems, financial planners, diagnostic systems for determining soybean pathology, and a meteorological expert system in China for determining areas in which to establish rubber tree orchards. Another area of application, akin to expert systems, is that of information retrieval.

CHAPTER: 4 SYNCHRONOUS GENERATORS

A "synchronous" generator runs at a constant speed and draws its excitation from a power source external or independent of the load or transmission network it is supplying. A synchronous generator has an exciter that enables the synchronous generator to produce its own “reactive" power and to also regulate its voltage. Synchronous generators can operate in parallel with the utility or in "stand-alone" or "island" mode. Synchronous generators require a speed reduction gear.

Customers worried about future blackouts and having increased power reliability should only consider cogeneration and regeneration power plants that have synchronous generators. Additionally, systems with synchronous generators can provide up to 100% of the facility's power, whereas induction generators can only supply about 1/3 of the facility's power requirements.

A synchronous machine is an ac rotating machine whose speed under steady state condition is proportional to the frequency of the current in its armature. The magnetic field created by the armature currents rotates at the same speed as that created by the field current on the rotor, which is rotating at the synchronous speed, and a steady torque results. Synchronous machines are commonly used as generators especially for large power systems, such as turbine generators and hydroelectric generators in the grid power supply. Because the rotor speed is proportional to the frequency of excitation, synchronous motors can be used in situations where constant speed drive is required. Since the reactive power generated by a synchronous machine can be adjusted by controlling the magnitude of the rotor field current, unloaded synchronous machines are also often installed in power systems solely for power factor correction or for control of reactive kVA flow. Such machines, known as synchronous condensers, may be more economical in the large sizes than static capacitors.

With power electronic variable voltage variable frequency (VVVF) power supplies, synchronous motors, especially those with permanent magnet rotors, are widely used for variable speed drives. If the stator excitation of a permanent magnet motor is controlled by its rotor position such that the stator field is always 90^0 (electrical) ahead of the rotor, the motor performance can be very close to the conventional brushed dc motors, which is very much favored for variable

speed drives. The rotor position can be either detected by using rotor position sensors or deduced from the induced emf in the stator windings. Since these types of motors do not need brushes, they are known as brushless dc motors.

4.1 SYNCHRONOUS MACHINE STRUCTURES:

4.1.1 Stator and Rotor:

The armature winding of a conventional synchronous machine is almost invariably on the stator and is usually a three phase winding. The field winding is usually on the rotor and excited by dc current, or permanent magnets. The dc power supply required for excitation usually is supplied through a dc generator known as exciter, which is often mounted on the same shaft as the synchronous machine. Various excitation systems using ac exciter and solid state rectifiers are used with large turbine generators.

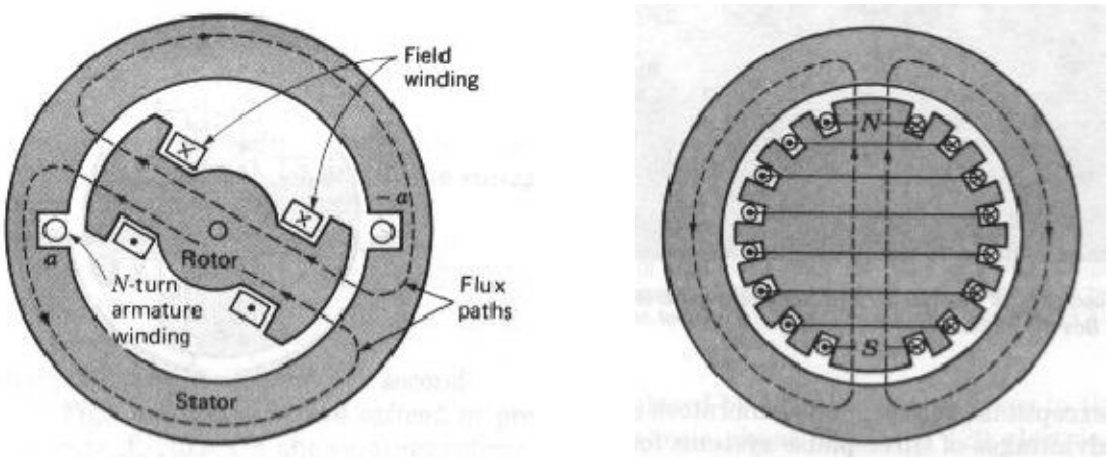


Figure**:** 4.1 Schematic Illustration of Synchronous Machines of
(a) Round or Cylindrical Rotor and (b) Salient Rotor Structures

There are two types of rotor structures: round or cylindrical rotor and salient pole rotor as illustrated schematically in the diagram below. Generally, round rotor structure is used for high speed synchronous machines, such as steam turbine generators, while salient pole structure is

used for low speed applications, such as hydroelectric generators. The pictures below show the stator and rotor of a hydroelectric generator and the rotor of a turbine generator.

4.1.2 Angle in Electrical and Mechanical Units:

Consider a synchronous machine with two magnetic poles. The idealized radial distribution of the air gap flux density is sinusoidal along the air gap. When the rotor rotates for one revolution, the induced emf, which is also sinusoidal, varies for one cycle as illustrated by the waveforms in the diagram below. If we measure the rotor position by physical or mechanical degrees or radians and the phase angles of the flux density and emf by **electrical degrees or radians**, in this case, it is ready to see that the angle measured in mechanical degrees or radians is equal to that measured in electrical degrees or radians, i.e.

$$\theta = \theta_m$$

where q is the angle in electrical degrees or radians and qm the mechanical angle.

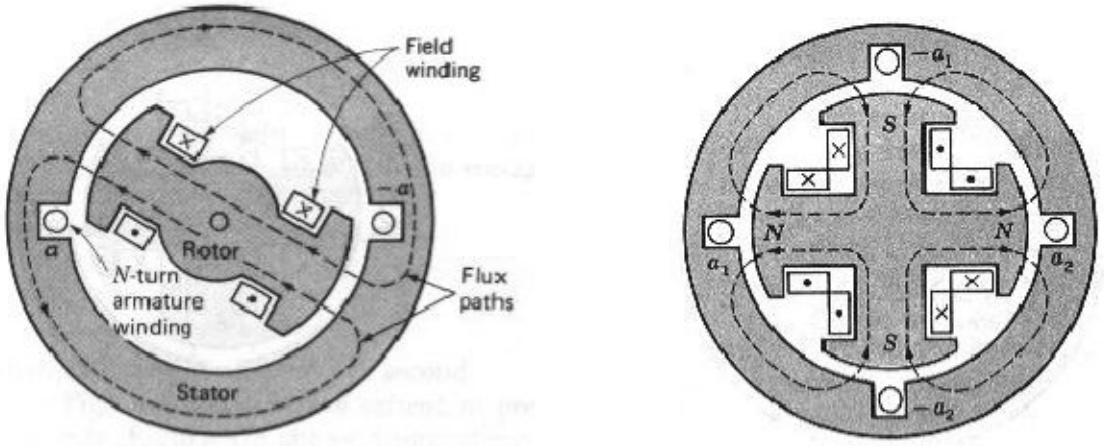


Figure: 4.2 Flux Density Distribution In Air Gap And Induced emf In The Phase Winding of a (a) Two Pole and (b) Four Pole Synchronous Machine

4.1.3 Rotating Magnetic Fields:

The magnetic field distribution of a distributed phase winding can be obtained by adding the fields generated by all the coils of the winding. The diagram below plots the profiles of mmf and field strength of a single coil in a uniform air gap. If the permeability of the iron is assumed to be

infinite, by Ampere's law, the mmf across each air gap would be Nia/2, where N is the number of turns of the coil and ia the current in the coil. The mmf distribution along the air gap is a square wave. Because of the uniform air gap, the spatial distribution of magnetic field strength is the same as that of mmf.

It can be shown analytically that the fundamental component is the major component when the square wave mmf is expanded into a Fourier Series, and it can written as When the field distributions of a number of distributed coils are combined, the resultant field distribution is close to a sine wave, as shown in the diagram in the next page. The fundamental component of the resultant mmf can be obtained by adding the fundamental components of these individual coils.

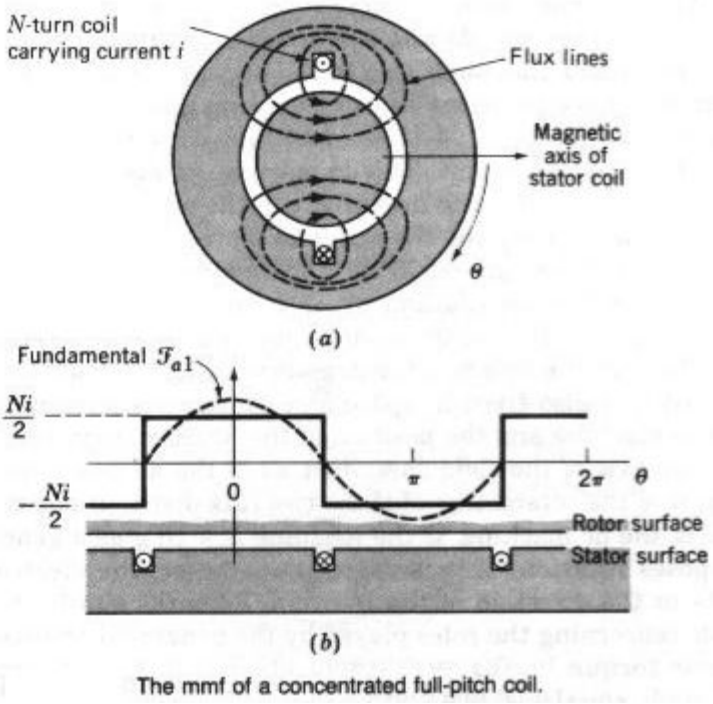


Figure: 4.3 Rotating Magnetic Field

4.2 3-PHASE GENERATOR (OR MOTOR) PRINCIPLES:

All 3-phase generators (or motors) use a rotating magnetic field. In the picture to the left we have installed three electromagnets around a circle. Each of the three magnets is connected to its own phase in the three phase electrical grid.

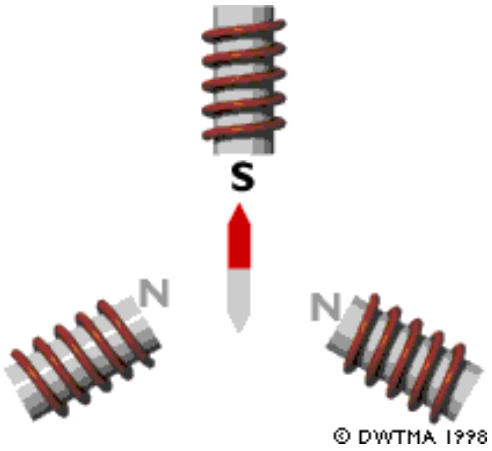


Figure: 4.4 3-Phases Generator

As you can see, each of the three electromagnets alternate between producing a South Pole and a North Pole towards the centre. The letters are shown in black when the magnetism is strong and in light grey when the magnetism is weak. The fluctuation in magnetism corresponds exactly to the fluctuation in voltage of each phase. When one phase is at its peak, the other two have the current running in the opposite direction, at half the voltage. Since the timing of current in the three magnets is one third of a cycle apart, the magnetic field will make one complete revolution per cycle.

The synchronous machine is one in which a.c. flows in the armature winding and D.C. is applied to he field winding. The armature winding is usually on the stator. Synchronous generators are usually rated in terms of the maximum kVA loads at the specified voltage and power factor, which they carry continuously without overheating. The main steady-state operating characteristics are:

I. Field current versus armature current

II. Terminal voltage versus armature current.

Consider a synchronous generator delivering power at constant frequency to a unity power factor (i.e. resistive) load. The curve showing the field current required to maintain rated terminal voltage as the constant power factor load is varied is known as the compounding curve. The compounding curve at any other power factor can also be determined. If the field current is held constant while the load varies, the terminal voltage will vary. Characteristic curves of terminal voltage can be plotted against armature current for any constant power factor load. The curve can

be drawn for one value of field current which is usually the value required to give rated terminal voltage at rated armature current. The variation of terminal voltage with load is due to the influence of armature reaction. When the power factor of the load is unity, the fall in voltage with increase of load is comparatively small. With an inductive load, the demagnetizing effect of armature e reaction causes the terminal voltage to fall much more rapidly. In many industrial installations, fluctuations of load are heavy. Due to rapid variations of load from instant to voltage also fluctuates considerably, because of the varying voltage drop in the armature circuit. To overcome this unsatisfactory feature, automatic voltage regulators are usually provided to maintain the generator voltage reasonably constant in spite of the fluctuating load. The voltage is increased when the load is high and decreased when the load comes down.

4.3 SYNCHRONOUS MOTOR OPERATION:

The compass needle (with the North Pole painted red) will follow the magnetic field exactly, and make one revolution per cycle. With a 50 Hz grid, the needle will make 50 revolutions per second, i.e. 50 times 60 = 3000 rpm (revolutions per minute). In the picture above, we have in fact managed to build what is called a 2-pole permanent magnet synchronous motor. The reason why it is called a synchronous motor, is that the magnet in the centre will rotate at a constant speed which is synchronous with (running exactly like the cycle in) the rotation of the magnetic field.

The reason why it is called a 2-pole motor is that it has one North and one South Pole. It may look like three poles to you, but in fact the compass needle feels the pull from the sum of the magnetic fields around its own magnetic field. So, if the magnet at the top is a strong South pole, the two magnets at the bottom will add up to a strong North pole.

The reason why it is called a permanent magnet motor is that the compass needle in the centre is a permanent magnet, not an electromagnet. (You could make a real motor by replacing the compass needle by a powerful permanent magnet or an electromagnet which maintains its magnetism through a coil (wound around an iron core) which is fed with direct current).

The setup with the three electromagnets is called the stator in the motor, because this part of the motor remains static (in the same place). The compass needle in the centre is called the rotor, obviously because it rotates.

4.4 SYNCHRONOUS GENERATOR OPERATION:

If you start forcing the magnet around (instead of letting the current from the grid move it), you will discover that it works like a generator, sending alternating current back into the grid. (You should have a more powerful magnet to produce much electricity). The more force (torque) you apply, the more electricity you generate, but the generator will still run at the same speed dictated by the frequency of the electrical grid.

You may disconnect the generator completely from the grid, and start your own private 3-phase electricity grid, hooking your lamps up to the three coils around the electromagnets. (Remember the principle of magnetic / electrical induction_from the reference manual section of this web site). If you disconnect the generator from the main grid, however, you will have to crank it at a constant rotational speed in order to produce alternating current with a constant frequency. Consequently, with this type of generator you will normally want to use an indirect grid connection of the generator.

In practice, permanent magnet synchronous generators are not used very much. There are several reasons for this. One reason is that permanent magnets tend to become demagnetized by working in the powerful magnetic fields inside a generator. Another reason is that powerful magnets (made of rare earth metals, e.g. Neodymium) are quite expensive, even if prices have dropped lately.

4.5 SYNCHRONOUS GENERATOR CAPABILITY LIMIT:

Synchronous generator capability limiters are as follows

1. MVA or armature current limit of generator: this depends on the cooling system of generator so that temperature rise in generator is limited to safe value. Depending on cooling system effectiveness and temperature limit for the insulation used in generator, MVA limit is decided.
2. MW limit: this is determined by the power output capacity of prime mover to which generator is connected.
3. Rotor angle limit: this is related to stability of generator which is synchronized to the grid. Ideally this could be 90 degree, but in practice this is limited to70 degree so as to have better stability margin in transient and dynamic condition. The generator falls out of synchronism in transient condition if rotor angle is close to 90 degree.

4. Rotor current limit: the field winding placed on rotor has got limited current carrying capacity, beyond which it may burn. So this limit issued. All these limiters make capability curve of generator within which the generators operates safely.

4.6 OBSERVATION:

Synchronous machines are commonly used as generator especially for large power system, such as turbine generator and hydroelectric generators in the grid power supply, because the rotor speed is proportional to the frequency of excitation, synchronous motors can be used in situation where constant speed drive is required.

When specifying, sizing, selecting a generator, we take many factors into consideration. Factor that relate to the prime mover must receive equal attention simultaneously. The specifications of synchronous generator in this case are as below.

1. Type - Turbo generator
2. RPM and frequency (187.5 rpm and 50 Hz)
3. Kilowatt rating and efficiency (2400kw and 94.7%)
4. Number of phase and power factor (3 & 0.8 lagging)

The experiment was conducted while alternator coupled to DC motor and field winding being excited. Armature terminal were kept open. The observation data obtained is given in Table 1.

Table 4.1: Observation data

S.N.	DC Motor armature terminal voltage (V)	DC Motor armature current (A)	Alternator field current If (A)	Alternator armature terminal voltage (V)
1	212	4.10	1.1	123
2	212	4.15	0.5	151
3	212	4.18	1.9	156
4	212	4.20	2.7	156
5	210	4.20	1.5	165
6	208	4.26	2.5	172

7	209	4.25	0.9	176
8	206	4.10	0.7	190
9	200	4.00	3.0	200
10.	200	4.10	0.3	220

While DC motor armature terminal voltage decreases, alternator terminal voltage increases monotonically. However the two currents viz. DC motor armature and alternator field current exhibit swing behavior.

CHAPTER: 5 FUZZY ENTROPY

5.1 DEFINITION:

In information theory, the information entropy is a measure of the uncertainty associated with a random variable. It quantifies the information contained in a message, usually in bits or bits/symbol. It is the minimum message length necessary to communicate information. This also represents an absolute limit on the best possible lossless compression of any communication: treating a message as a series of symbols, the shortest possible representation to transmit the message is the Shannon entropy in bits/symbol multiplied by the number of symbols in the original message.

A fair coin has entropy of one bit. However, if the coin is not fair, then the uncertainty is lower (if asked to bet on the next outcome, we would bet preferentially on the most frequent result), and thus the Shannon entropy is lower. A long string of repeating characters has entropy of 0, since every character is predictable. The entropy of English text is between 1.0 and 1.5 bits per letter, or as low as 0.6 to 1.3 bits per letter, according to estimates by Shannon based on human experiments.

Equivalently, the Shannon entropy is a measure of the average information content the recipient is missing when he does not know the value of the random variable.

5.2 ILLUSTRATIVE EXAMPLE:

Consider tossing a coin with known, not necessarily fair, probabilities of coming up heads or tails. The entropy of the unknown result of the next toss of the coin is maximized if the coin is fair (that is, if heads and tails both have equal probabilities 1/2). This is the situation of maximum uncertainty as it is most difficult to predict the outcome of the next toss; the result of each toss of the coin delivers a full 1 bit of information. However, if we know the coin is not fair, but comes up heads or tails with probabilities p and q, then there is less uncertainty. Every time, one side is more likely to come up than the other. The reduced uncertainty is quantified in lower entropy. On average each toss of the coin delivers less than a full 1 bit of information.

The extreme case is that of a double-headed coin which never comes up tails. Then there is no uncertainty. The entropy is zero: each toss of the coin delivers no information. Here, then, a seven is the result with the highest entropy (i.e. probability), and a 2 ("snake eyes") or a 12 ("boxcars") have the lowest entropy. The entropy is actually k ln (# combinations), where k is called Boltzmann's constant and ln means the natural logarithm.

Now we consider a "box" with two white marbles and two black marbles inside it. The box is made so that exactly two of the marbles are always on the left hand side and two are always on the right hand side of the box. In the case shown to the right, both white marbles are on the left side of the box and both black marbles are on the right side of the box. There is only one combination of marbles that gives this arrangement.

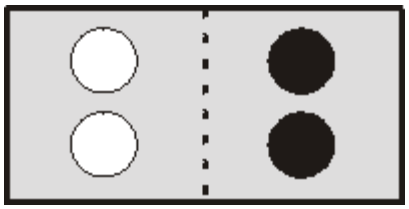


Figure:5.1 Marbles

Imagine that we shake the box, and how the marbles distribute themselves is random. Now we have a white marble and a black marble on the left, and a white and black marble on the right. Call the black marbles *B1* and *B2*, and the white ones *W1* and *W2*. Then for this arrangement we could have B1, W1 on the left and B2, W2 on the right. We could also have B1, W2 on the left and B2, W1 on the right; or B2W1 on the left and B1, W2 on the right; or B2, W2 on the left and B1, W1 on the right. Thus there are four combinations that give this arrangement of the marbles.

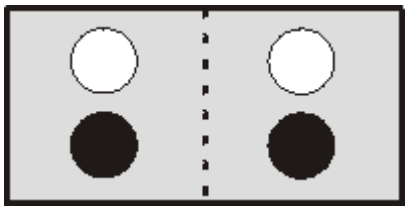


Figure:5.2Marbles Distribution

Finally, we show the single arrangement with both black marbles on the left and both white ones on the right.

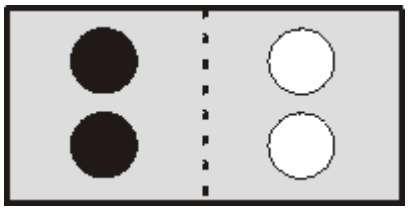


Figure:5.3Marbles Distribution

For the four marbles in a box that we just considered, there are a total of six equally likely combinations. The one with the highest probability (4/6=67%) has a black marble and a white marble on each side of the box. This is therefore the state with the highest entropy. The graph to the right shows the total number of combinations.

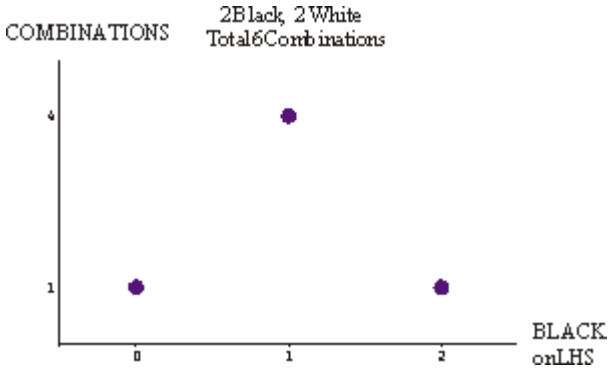


Figure:5.4 Combinations with 2W, 2B

For six marbles, three black and three white, the equivalent plot is shown to the right.

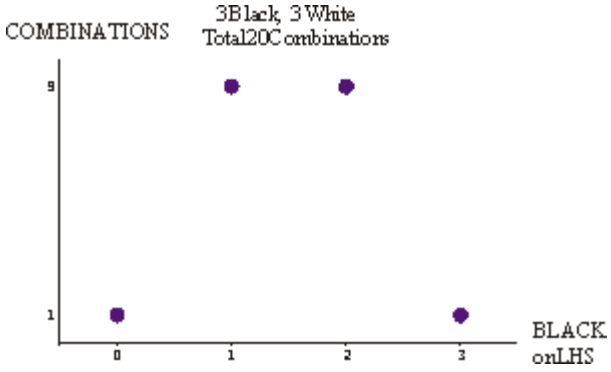


Figure:5.5 Combinations with 3W, 3B

Now we jump to a total of 12 marbles, six black and six white. (Here I did not find the number of combinations by brute force counting: I used some mathematics.)

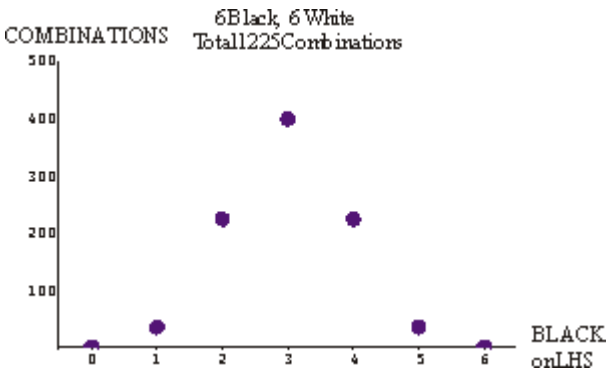


Figure: 5.6 Combinations with 6W, 6B

Here is the equivalent plot for a total of 100 marbles. The thing to notice is that the most probable result remains 50% black on the left, 50% black on the right, 50% white on the left and 50% white on the right. However, as the total number of marbles increases, the "peak" becomes sharper. This means that the probability of getting a result far away from 50% gets smaller and smaller the greater the number of marbles. You may wish to know that the total number of combinations for this case is about 1 followed by 29 zeroes before the decimal point.

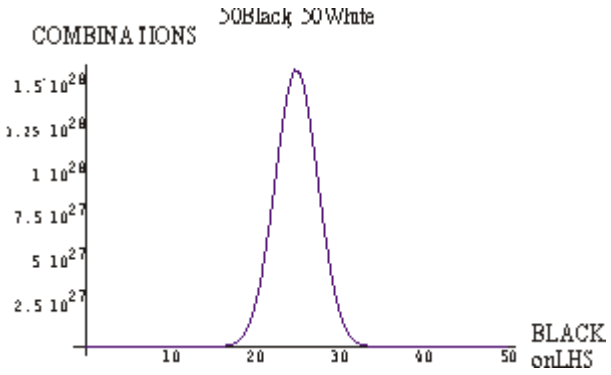


Figure:5.7Combinations with 50W, 50B

For a total of 400 marbles, the point we just made about the peak becoming narrower becomes even more obvious. Here the total number of combinations is about 1 followed by 119 zeros before the decimal point.

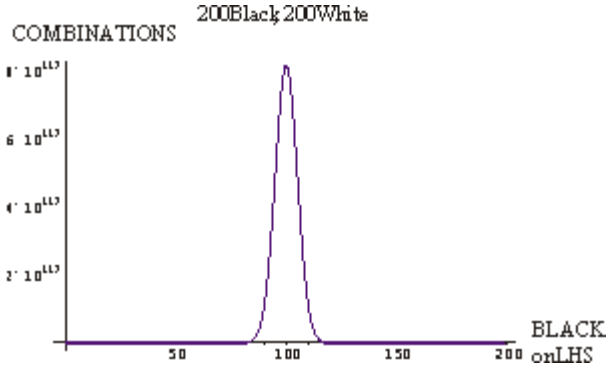


Figure:5.8Combinations with 200W, 200B

The conclusion from all this is that the most probable highest entropy result is the one in which equal numbers of white and black marbles occur on both sides of the box, and that as the total number of marbles increases the probability of getting a distribution very far from this most probable result goes down. Imagine the case above with a total of 200 black marbles and 200 white marbles. If all the white marbles are on the left hand side of the box and all the black ones are on the right hand side of the box, we say that the marbles are very ordered. The state where all the all black marbles on the left and all the white ones are on the right are similarly ordered.

If we have 100 black marbles on the left, 100 on the right, 100 white marbles on the left and 100 on the right, we say the system is very disordered, which is another way of saying that the two sides of the box are undifferentiated. Note that this is also the most probable state. Thus we are led to our second equivalent definition of entropy.

5.3 MEASURE OF FUZZY ENTROPY:

The entropy is a measure of the disorder of a system. It is easy to get confused because this definition is the measure of the lack of order in a system. Usually we think of quantities such as mass, energy, etc. as measuring the amount of that something; here we are measuring the lack of that something. Some people like to talk about the negentropy, which is the negative of the entropy; it measures the order of the system.

It is a measure of the amount of uncertainty of fuzzy set. Fuzzy entropy discriminates the best number of intervals for the quantitative attribute. The fuzzy entropy for each interval is defined as below.

1. Let X = {x1, x2 …xn} be a universal set with elements xi distributed in a pattern space, where I = 1, 2…n.
2. Let A be a fuzzy set defined on an interval of pattern space, which contains k element (k<n). The mapped membership degree of the element xi with the fuzzy set A is denoted by μA (xi).
3. Let C1, C2 …Ck represent k classes into which the n elements are divided.

4. Let Sf(xn) denote a set of element of class j on the universal set X. it is a subset of universal set X.
5. The match degree Dj with the fuzzy set A for the elements of class j in an interval, where j = 1,2……..k is defined as

$$Dj = \sum_{X \in Sj(xn)} \mu A(r) \Big/ \sum_{x \in S} \mu A(r) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (1)$$

6. The fuzzy entropy FEj (A) = - Dj log2 Dj

7. The fuzzy entropy FE (A) on the universal set x for the element within an interval is defined as

$$FE(A) = \sum_{j=1}^{k} FECj(A) \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

CHAPTER: 6 PROBLEM FORMULATION

6.1 FUZZIFICATION:

In complex nonlinear knowledge discovery problems some suitable nonlinear fuzzy sets like sigmoid, trapezoidal etc can be chosen.

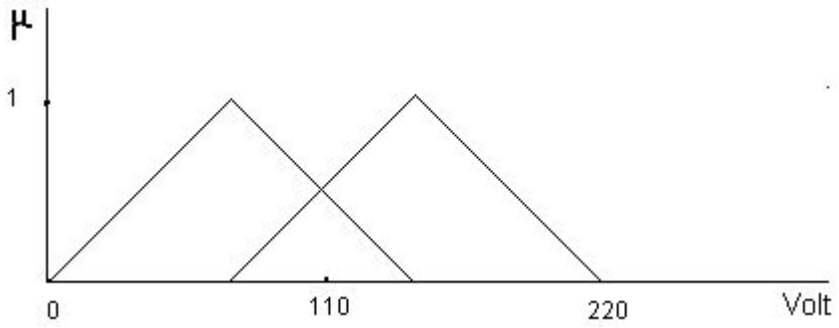


Figure: 6.1 Data Classification "1" For Assigning Membership Values

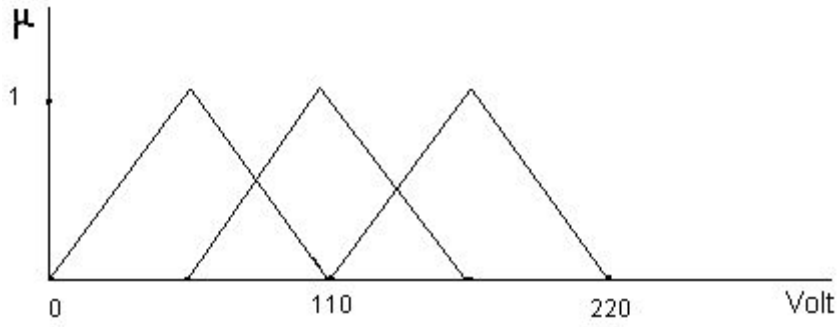


Figure 6.2: Data Classification "2" For Assigning Membership Values

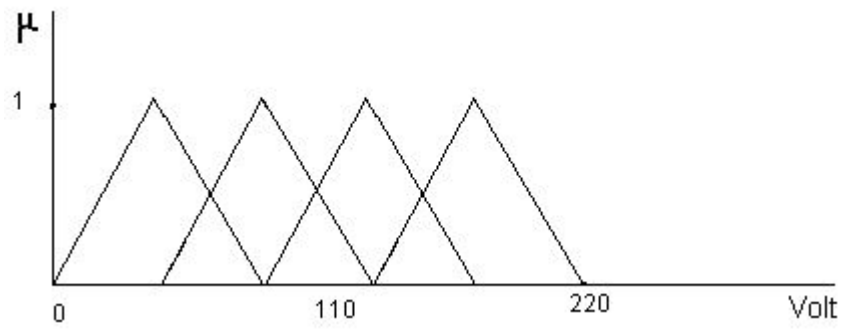


Figure 6.3: Data Classification "3" For Assigning Membership Values

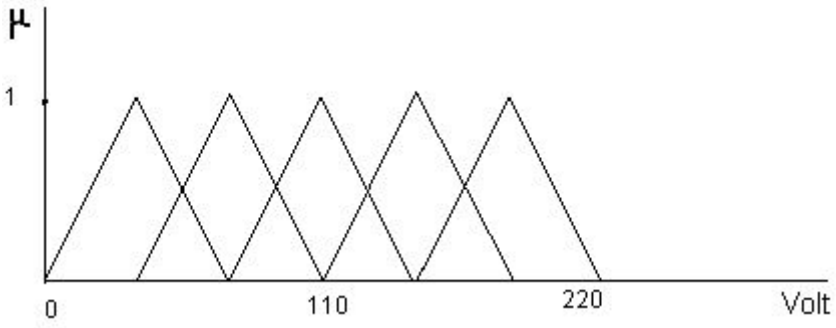


Figure 6.4: Data Classification "4" For Assigning Membership Values

6.2 ALGORITHM:

We have used symmetrical triangular form of membership function to represent fuzzy set for the complete data as shown in table 1.The optimum value on the basis of above membership function values is obtained as below.

Optimum value = $\sum$ If*μ / $\sum \mu$ …… (1)

The developed algorithm for correct membership value on the basis of Fuzzy Entropy in this case, is given below.

Step 1: Input: An incomplete quantitative data set in which the number of data and those values are given.

Step 2: Output: Assign a membership value function and find optimum value among those.

Step 3: Nd = number of data given.

V [I] =values of given data.

Rmax = Maximum value of given dataset.

Rmin = Minimum value of given dataset.

T = number of triangle between Rmin and Rmax.

Initialize

T = 2

Value of Fuzzy Entropy FE (previous) = 0

Value of Fuzzy Entropy FE (present) = 0

Value of optimum data, Vopt (previous) = 0

Value of optimum data, Vopt (present) = 0

Step 4:

Do

{

```
FE (previous) = FE (present)

FE (present)   = 0

Initialize the match degree D [j] = 0

                    Numerator Nr [j] = 0

                     Denominator Dr = 0



For (I =0; I<= Nd; I++)

 {


  For (J=0; J<= T; J++)

   {

   If (Rmin+ (Rmax*(J-1)/ (T+1)) < V [I] <= Rmin+ (Rmax*(J-1)/ (T+1)))

   {

   Nr [J] =Nr [J]-(V [I]*(T+1)/ (Rmax-Rmin)) + ((T+1)*(Rmin+ (Rmax-Rmin)*(J+1)/ (T+1))/
 (Rmax-Rmin));

   Nr [j+1] =Nr [j+1] + (V [I]*(T+1)/ (Rmax-Rmin))-((T+1)*(Rmin+ (Rmax-Rmin)*J/ (T+1))/
 (Rmax-Rmin));

   }

  }


  For (J=0; J<= 2.T; J=J+2)
```

```
 {
  If ((Rmin+ (Rmax-Rmin)*J/ ((T+1)*2)) <V [I] <= (Rmin+ (Rmax-Rmin)*(J+1)/
((T+1)*2)))
    {
     Msp=-(V [I]*(T+1)/ (Rmax-Rmin)) + ((T+1)*(Rmin+ (Rmax-Rmin)*(J+2)/ ((T+1)*2))/
(Rmax-Rmin));
     Nropt=Nropt + Msp*V [I];
     }

   If ((Rmin+(Rmax-Rmin)*(j+1)/((nt+1)*2))<V[I]<=(Rmin+(Rmax-
Rmin)*(j+2)/((nt+1)*2)))
    {
      Msp= (V [I]*(T+1)/ (Rmax-Rmin))-((T+1)*(Rmin+ (Rmax-Rmin)*(j+2)/ ((T+1)*2))/
(Rmax-Rmin));
      Nropt=Nropt + Msp*V [I];
    }

  Dr=Dr+Msp
 }
}

For (J=1; J<=T; J++)
```

{

D [J] = Nr [J] / Dr

FE (present) = FE (present) – D [J]. log2 D[J]

}

Vopt (previous) = Vopt (present)

Vopt (present) = Nropt/Dr

T++

} While (FE (present) > FE (previous))

Optimum value of given dataset = Vopt (previous)

Step5: T = T-2

T is number of symmetrical triangle exist between 0 and Rmax. Use this Fuzzy membership function value for finding missing values. Defuzzify the function for finding the optimum value.

6.3 ENTROPY CHECKING:

Table 6.1: Fuzzy Entropy (FE) For Desired Optimum Value of Alternator Armature Terminal Voltage.

Serial NO	No of class between 0 and max. input (based on membership function)	Total Fuzzy Entropy (FE)

1	2	0.869630672
2	3	1.424813196
3	4	2.017413200
4	5	2.159269261

Here Fuzzy Entropy (FE) is minimum when number of class interval is 2. So we will chose membership value according to this interval.

Now putting the values in Eq. 1, we get

Optimum value

$$= \underline{(123*2.7 + 151*3.7 + 156*3.5 + 156*3.5 + 165*3 + 172*2.6 + 176*2.4 + 190*1.6 + 200*1.1 + 220*0)/4}$$

$$(2.7+3.7+3.5+3.5+3+2.6+2.4+1.6+1.1+0)/4$$

= 160.64 volt

From above, the optimum value for the Alternator armature terminal voltage is equal to 160.64 volt.

CHAPTER: 7 SIMULATION AND TESTING

The above algorithm for Fuzzy Entropy has been coded in a higher-level language. A set of 10 incomplete data pertaining to the Synchronous generator has been fed to the code and the simulation results have been obtained. The testing results are given in table 7.1 below.

Table 7.1: For Calculating The Accuracy Of This Technique

Sl. no.	Data range	Desired data value VD (average value of input data's)	Computed value from this algorithm VC	Error = (VC-VD)100 / VC (In %)
1	123,151,156,156,165,172,176,190,200,220	170.9	160.64	+6.01
2	127,135,158,170,150,186,210,202,166,156	166.0	158.03	+4.8
3	148,162,160,188,200,195,190,172,170,180	176.5	170.85	+3.2
4	155,160,162,190,176,152,158,205,180,150	168.8	160.02	+5.2
5	140,148,173,200,210,205,164,157,151,180	172.8	163.98	+5.1
6	154,158,191,176,200,168,165,158,195,188	175.3	182.48	-4.1
7	160,178,192,144,146,198,200,151,158,149	167.6	157.37	+6.1
8	168,172,165,161,183,195,193,180,175,200	179.2	184.39	-2.9
9	155,159,163,168,185,191,186,205,207,196	181.5	187.85	-3.5
10	168,153,156,178,182,191,202,195,180,175	178.0	172.48	+3.1

Average error = ±4.40%

CHAPTER: 8 RESULTS AND DISCUSSIONS

Maximization and minimization of simplest measures of entropy is subjected to $\sum \mu_i(X_i) = k$. we have found it convenient for this purpose to take all the sets in the standard forms when each $\mu_A(X_i) \leq ½$. We have shown that maximum entropy is increasing continuous function of which is differentiable everywhere. Minimum entropy is continuous function of k which is not differentiable at some points. For a fuzzy set, we may not given the values of all of $\mu_A(X_1)$, $\mu_A(X_2)\ldots\ \mu_A(X_n)$, but we may give some partial information about these in the form of equality or inequality relation between the values of these. We have given a method for measuring the information provided by each of these pieces of knowledge. This knowledge will change if some prior information based on intuition or experience is available about the possible values of these membership functions. We have considered here how this information is modified in this case. Fuzzy entropy measures uncertainty due to fuzziness of information, while probabilistic entropy measures uncertainty due to the information being available in terms of a probability distribution only.

The results as given in table 7.1 have been checked with their actual values. The average error for the data set comprising of ten uncertain values for this case study comes out to be $\pm 4.40\%$. Here we are observing that computed value is deviating very less from observed value, but at this computed value fuzzy entropy is minimum so this technique gives the better option to select the value for further processing.

CONCLUSIONS AND FUTURE SCOPE

In maximizing and minimizing fuzzy entropy, the fuzziness of the fuzzy vector does not change when any component $\mu_A(X_i)$ is replaced by $1-\mu_A(X_i)$. Fuzzy entropy measures uncertainty due to fuzziness of information, while probabilistic entropy measures uncertainty due to the information being available in terms of a probability distribution only.

Measures of fuzzy entropy and probabilistic entropy have a great deal in common and the knowledge of measures of probabilistic entropy may be used to enrich the literature of fuzzy measures and vice-versa. Since measures of inaccuracy and information improvement are defined in terms of measures of directed divergence, we can also obtain a new class of measures of inaccuracy and information improvement.

Fuzzy entropy is one of the best knowledge discovery methods as soon in the paper data classification is one of the important research areas. Fuzzy entropy is very efficient in handling incomplete qualitative and nominal data. The results of the proposed algorithm demonstrate an error of $\pm 4.40\%$. The knowledge discovery based on fuzzy entropy has been made further efficient by using a supervised by learning based on genetic programming. Genetic programming is one of the best available methods for generating the data classifiers.

REFERENCES

1. Yizong Cheng, 1994, Fuzzy Clustering as Blurring. Fuzzy Systems,. IEEE World Congress on Computational Intelligence., Proceedings of the Third IEEE Conference, 1830-1834 vol.3

2. M. Delgado , A. Gomez-Skarmeta , M.A. Vila , 1995. Hierarchical Clustering to validate Fuzzy Clustering. Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE International Conference, 1807-1812 vol.4.

3. Mika Sato And Yoshiharu Sato ,1995. Fuzzy Clustering Model for Fuzzy Data Fuzzy Systems, 1995. International Joint Conference of the Fourth IEEE International Conference on Fuzzy Systems and The Second International Fuzzy Engineering Symposium., Proceedings of 1995 IEEE International Conference On page(s): 2123-2128 vol.4

4. Jianhua Chen and Jaspal Sabharwal, 1996 Intelligent pH Control Using Fuzzy Linear Invariant Clustering, Proceedings of the 28th Southeastern Symposium on System Theory (SSST '96), page(s): 514 -520

5. F. Klawonn, Member, rEEE, and R. Kruse, 1996. Automatic Generation of Fuzzy Controllers by Fuzzy Clustering, Fuzzy Systems, 1996., Proceedings of the Fifth IEEE International Conference, page(s): 2053-2058 vol.3

6. Chen, J. Kundu, S., 1996. Fuzzy control system design by fuzzy clustering and self-organization, Fuzzy Information Processing Society, 1996. NAFIPS. 1996 Biennial Conference of the North American On page(s): 456-460

7. Chih-Hsiu Wei and Chin-Shyurng Fahn, 1996 A Distributed Approach to Fuzzy Clustering by Genetic Algorithms, Fuzzy Systems Symposium, 1996. 'Soft Computing in Intelligent Systems and Information Processing'., Proceedings of the 1996 Asian, page(s): 350-357

8. Linkens, D.A. Min-You Chen, 1998, Hierarchical fuzzy clustering based on self-organising networks, Fuzzy Systems Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on page(s): 1406-1410 vol.2

9. Miyamoto, S, 1998 An overview and new methods in fuzzy clustering. Knowledge-Based Intelligent Electronic Systems, 1998. Proceedings KES '98. 1998 Second International Conference on page(s): 33-40 vol.1

10. Kreinovich, V. Nguyen, H.T. Starks, S.A. Yeung Yam, 1998. Decision making based on satellite images: optimal fuzzy clustering approach. Decision and Control, 1998. Proceedings of the 37th IEEE Conference on page(s): 4246-4251 vol.4

11. Tai Wai Cheng Dmitry B. Goldgof and Lawrence O. Hall, 1998, Fast fuzzy clustering, Fuzzy Sets and Systems Volume 93 , Issue 1 (January 1998)

12. Altman, D., 1999. Efficient fuzzy clustering of multi-spectral images. Geoscience and Remote Sensing Symposium, 1999. IGARSS '99 Proceedings. IEEE 1999 International On page(s): 1594-1596 vol.3

13. A.K. Jain, M.N. Murty, P.J. Flynn,1999, Data clustering: a review, ACM Computing Surveys (CSUR) Volume 31 , Issue 3, Pages: 264 – 323.

14. Nas, T. and Mevik, B.-H. (1999); The Flexibility of Fuzzy Clustering Illustrated by Examples; Journal of Chemometrics 13(3—4), 435—444.

15. P.J. COSTA BRANCO and J.A. DENTE, 1999. Noise Effects in Fuzzy Modeling Systems:Three Case Studies, Computational Intelligence and Applications, pp. 103-108.

16. Keller, A., Klawonn, F., 1999, Context sensitive fuzzy clustering, Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American On page(s): 347-351

17. Geva, A.B., 1999. Non-stationary time-series prediction using fuzzy clustering, Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American On page(s): 413-417

18. Russell, S. Lodwick, W. ,1999. Fuzzy clustering in data mining for telco database marketing campaigns. Fuzzy Information Processing Society, 1999. NAFIPS. 18th International Conference of the North American On page(s): 720-726

19. Steven Schockaert, Martine De Cock, and Etienne E. Kerre,2000 Automatic Acquisition of Fuzzy Footprints (http://www.fuzzy.ugent.be)Schneider, A., 2000. Weighted possibilistic c-means clustering algorithms, Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on page(s): 176-180 vol.1

20. Kraft, D.H., Chen, J. Mikulcic, A. , 2000. Combining fuzzy clustering and fuzzy inferencing in information retrieval. Fuzzy Systems, 2000. FUZZ IEEE 2000. The Ninth IEEE International Conference on page(s): 375-380 vol.1

21. N. Mac Parthalain, R. Jensen and Q. Shen. Fuzzy entropy-assisted fuzzy-rough feature selection. Proceedings of the 15th International Conference on Fuzzy Systems (FUZZ-IEEE'06). 2006.

22. J.W. Grzymala-Busse and M. Hu, “A comparison of several approaches to missing attribute value in data mining,” in Proc. Of Second International conference on rough Sets and current trends In Computing, RSCTC2000, pp.378-385.

23. T-P Hong, L-H Tseng and B-C. Chien," Learning Fuzzy rules from incomplete quantitative data by rough sets'" in Proc. Of the 2002 IEEE International Conference on Fuzzy System, pp.1438-1443.

24. K.M. Faraoun, and A. Boukelif "Genetic Programming Approach for Multi-Category Pattern Classification Applied to Network Intrusions Detection" in International Journal of Computational Intelligence Volume .3.

25. M. Bramrier and W. Banzhaf, "A comparison of linear genetic programming and neural networks in medical data mining " IEEE trans. On Evolutionary Computation, Vol.5, No. 1 Feb 2001, pp. 17-26.

26. Ralf Mikut, Jens Jäkel, Lutz Gröll, "Interpretability issues in data-based learning of fuzzy systems" in Proc. Of the 2004 IEEE International Conference on Fuzzy System, pp.1428-1433

27. R. Jensen and Q. Shen. Fuzzy-Rough Sets Assisted Attribute Selection. IEEE Transactions on Fuzzy Systems, vol. 15, no. 1, pp. 73-89, 2007.

28. Włodzisław Duch "Similarity-based methods: a general framework for classification, approximation and association" in Proc. Of Second International conference on fuzzy Sets and current trends In Computing, FSCTC2000, pp.245-252

29. Richard E. Haskell "Neuro-Fuzzy Classification and Regression Trees" in NAFIPS. 1996 Biennial Conference of the North American On page(s): 456-460.

APPENDIX 1 C/C++ CODE

```
/* program to implement efficient knowledge discovery using fuzzy entropy and finding
optimum path*/

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<string.h>

void main()

{

int  nd, i, j, nt=2;

float v[50], feprevious, fepresent, rmax, rmin, nr[50], d[50], dr=0.0, nropt=0.0, voptpresent,
voptprevious, msp, x, y, z, sum;

clrscr();

printf("\nenter the total no of data, you want to give ");

scanf("%d",&nd);


printf("\n\nnow enter the values of data one by one\n\n");

for(i=0;i<nd;i++)

        scanf("%f",&v[i]);


rmin = v[0];

rmax = v[0];

    z = nt++;

for(i=0;i<nd;i++)
```

```
    {
     if(rmax<v[i])
     rmax = v[i];
     if (rmin>v[i])
     rmin = v[i];
     sum = sum + v[i];
    }
do
   {
    for(i=0;i<50;i++)
          {
           d[i]=0.0;
           nr[i]=0.0;
          }
    Feprevious = fepresent;
    Fepresent = 0.0;
    Nropt = 0.0;
    for(i=0;i<nd;i++)
          {
           for(j=0;j<=nt;j++)
              {
               if(rmin+((rmax-rmin)*j/(nt+1))<=v[i]<(rmin+(rmax-rmin)*(j+1)/(nt+1)))
                 {
                    nr[j] = nr[j]-(v[i]*(nt+1)/(rmax-rmin))+((nt+1)*(rmin+(rmax-
                    rmin)*(j+1)/(nt+1))/(rmax-rmin));
```

```
            nr[j+1] = nr[j+1]+v[i]*(nt+1)/(rmax-rmin)-((nt+1)*(rmin+(rmax-
            rmin)*j/(nt+1))/(rmax-rmin));

          }

        }


      for(j=0;j<=2*nt;j++)

        {

        if((rmin+(rmax-rmin)*j/((nt+1)*2))<v[i]<=(rmin+(rmax-rmin)*(j+1)/((nt+1)*2)))

          {

            msp=-(v[i]*(nt+1)/(rmax-rmin))+((nt+1)*(rmin+(rmax-
            rmin)*(j+2)/((nt+1)*2))/(rmax-rmin));

            nropt=nropt+msp*v[i];

          }

        if((rmin+(rmax-rmin)*(j+1)/((nt+1)*2))<v[i]<=(rmin+(rmax-rmin)*(j+2)/((nt+1)*2)))

          {

            msp=(v[i]*(nt+1)/(rmax-rmin))-((nt+1)*(rmin+(rmax-
            rmin)*(j+2)/((nt+1)*2))/(rmax-rmin));

            nropt=nropt+msp*v[i];

          }

        dr=dr+msp;

        }


      }


for(j=1;j<=nt;j++)

      {
```

```
        d[j]=(nr[j]/dr);
        fepresent = fepresent-(nt*j);
       }
  voptprevious=voptpresent;
  voptpresent=nropt/dr;
  nt++;
 }while(fepresent>feprevious);
 Sum = (sum/nd)*(1+(z/100));
 Voptpresent = sum;
 printf("\n\n the optimum value of given data= %f",voptpresent);
 getch();
}
```

APPENDIX 2 PUBLICATIONS FROM THIS RESEARCH WORK

- "An efficient knowledge discovery algorithm using fuzzy entropy in synchronous generator: a case study", has been accepted for the **IMECS (International Multiconference of Engineers and Computer Scientists) 2007 Hong Kong.** The paper was considered for the **Best Paper Award** in its corresponding workshop.

APPENDIX 3 BRIEF BIODATA OF THE RESEARCHER

AMIT AGRAWAL
Bharat Med. Agencies, Chhoti Bazar, Banda-210001, India
Email: aa_shalu@rediffmail.com
Phone: +91-9815953304

EDUCATION

M.E., Electronic Instrumentation and Control, T.I.E.T, Patiala (2006-2008) with C.G.P.A of 8.76

- **Thesis**: Design and development of an algorithm for fuzzy entropy
- **Language**: C/C++

B.Tech., Electronics and communication, Madan Mohan Malviya Engineering college, with 70%, U.P.T.U. Lucknow.

- **Project Title:** Data Acquisition System Interface To PC Classifier.
- **Advisor**: Dr R.K. Chauhan (M.M.M. Engg. College, GKP)
- **Language**: - C/C++, Matlab, VB6.0
- **Summary**: This system has been designed for those who want to analyze any type of signal with in few second. Here we implement a circuit model of DAS, which converts analog signal into digital signal compatible PC's parallel port.

G.I.C. Banda, **12th** (Graduated 1999) with 78% aggregate.

D.A.V. Inter College Banda, **10th (**Graduated 1997) with 80% aggregate.

PROJECT WORK (M. E)

Communication Network: Design and development of an algorithm for data encryption in multipath routing of communication network.

Artificial Intelligence: Detection of acoustic variability (speech recognition) of digits zero to nine using back propagation algorithm in artificial neural network (ANN).
An Efficient Knowledge Discovery: Implementation of algorithm based on Fuzzy Entropy.

PROJECT WORK (B.Tech.)

Process Controller: Designed a PC based level controller for controlling the water level in a process control loop.
Embedded System: Built a temperature and fan monitoring and control unit for the PC using microcontroller.
Analog Circuit Design: Built a **component tester** for various electronic circuits, to verify the connectivity in the circuit.
Microcontroller based model: Interfacing of graphical **LCD** (liquid crystal display) with PC using microcontroller.

TRAINING & WORK EXPERIENCE

- Aug-sept 2001: 'C' language course with 'A' grade from SVIIT GORAKHPUR
- Aug 2002: B.S.N.L. Banda. (About basic idea of telephone system i.e. Switching, Transmission system and Internet.)
- June-July 2003: Programmed on 'VHDL Programming' in DOEACC center Gorakhpur.
- One month teaching in K.C.N.I.T. Engineering College Banda (U.P.)
- Three month work in Bhushan Steel & Strips Ltd 23, Site-IV Sahibabad Indl. Area, Distt. Ghaziabad-201010 (U.P.).

COMPUTER SKILLS

Software: - Introduction Lab View 6.0 and Keil
Languages: - Introduction to C/C++, Mat Lab 6.0, LISP, and assembly
Platform: - Windows98 / XP/vista

PAPERS PUBLISHED

"International Multi Conference of Engineers and Computer Scientists, Hong Kong 2007". Also nominated for the **Best Paper Award** in the conference.

Topic: An efficient knowledge discovery algorithm using fuzzy entropy in synchronous generator: a case study.

AWARDS

- National scholarship holder.
- Rank-2 at district level in matriculation.
- Rank-1 at district level in intermediate.
- A.I.R. In IIT-pre: 1461 and A.I.R. In IIT-mains: 3872
- Qualified gate-2006, score: 328

REFERENCES

Dr. Yaduvir Singh, Dept of electrical and instrumentation, T.U., Patiala

Mrs Gagandeep Kaur, Dept of electrical and instrumentation, T.U., Patiala

Date: **(AMIT AGRAWAL)**