A

REPORT ON

APPLICATIONS OF FUZZY LOGIC IN

COMPUTER ENGINEERING

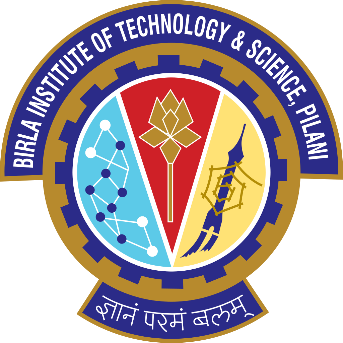
BY

VAISHNAVI BHARGAVA 2015B4A70777P

RASHMI LATHI 2015B4A70619P

DEEPIKA POONIA 2015B4TS0968P

AT



ACKNOWLEDGEMENT

The presented report is a component of the course Fuzzy Logic and Applications. This project, along with the course, has been a great learning experience for us. To all the people who made that possible, we extend our heartfelt gratitude and appreciation.

Course Fuzzy Logic is itself very interesting and there are many more fields to explore in which fuzzy has its application.

We are thankful to Dr. Chandra Shekhar for keeping this project in our course which increased our understanding and helped us applying it practically in different fields.

We are thankful to BITS for keeping this course in our curriculum and more importantly Library and IPC to provide us infinite access to the books and internet.

**INDEX**

1. Introduction
2. Fuzzy Logic in Different Branches of Computer Engineering
   1. Image Processing
   2. Data Mining
   3. Information Retrieval
   4. Cryptography
   5. Machine Learning
   6. Computer Security and Forensics
3. Fuzzy Logic for Image Processing
   1. Introduction
   2. Image Edge Detection
   3. Steps for Edge Detection
4. Fuzzy Logic in Computer Security and Forensics
   1. Introduction
   2. Fuzzy Logic Control
   3. Intrusion Detection

**Introduction**

Fuzzy logic is important tool for many different applications ranging from the control of engineering system to artificial intelligence. Practical applications of fuzzy logic have a unique set of problems. Fuzzy logic is applied in the design of systems, to make use of human knowledge and experience. It is a daunting task without facing engineering problems of real world systems.

The first step is to build a fuzzy model from the knowledge available for the problem domain. Designing of the fuzzy solution is comprised in this phase. After the design and prototyping phase, the solution is integrated into the production environment. It is the system which provides the fuzzy logic based algorithm with inputs and applies the output of the fuzzy logic solution.

The integration phase may involve a series of problems, depending on the target system. Sometimes, the prototyping and the design of the fuzzy logic based algorithm can be done directly and ported to the final system.

Even from the high level view of the steps involved in the development of the fuzzy logic solution, it can be seen that it involves steps which slow down the development and add a significant learning curve to the process. This hinders the adoption of fuzzy solution in different fields.

Fuzzy logic based approach allows the developers to focus on the decision logic of the algorithm. Also, the architecture of a fuzzy logic approach requires that an algorithm for a specific problem is built with a bottom up design. Here all sub steps of the decision algorithm are described and composed to reach the final solution.

There is a risk involved when designing a project and deciding to use fuzzy logic. This includes the previously mentioned phase of designing of the fuzzy logic solution for a specific algorithm. Modelling a domain specific problem in pseudo code is easier than achieving the same in a fuzzy logic description. It is highly unlikely that the domain expert has knowledge is/or an expert in fuzzy logic as well. It also introduces further difficulties in the maintenance of the algorithm implementation.

The principal idea employed by fuzzy logic is to allow for a partially ordered scale of truth values, called also truth degrees, which contains the values representing false and true but possibly also other, intermediary truth degrees. That is, the two element set {0, 1} of truth values of classical logic, where 0 and 1 represent false and true, respectively, is replaced in fuzzy logic by a partially ordered scale of truth degrees with the smallest degree being 0 and the largest one being 1. This is known as the graded approach.

**Fuzzy Logic in Different Branches of Computer Engineering**

* **Image Processing**

Image processing is a prominent area that supports applications in different fields, such as medicine, astronomy, national security, autonomous systems, product quality, industrial applications, etc.  Some algorithms developed under the scheme of traditional logic have crisp approach. As a consequence, information uncertainty and subjectivity are difficult to model in these algorithms.

A better structure to model a real system - the human visual system, must incorporate tools to model better subjective concepts and must be able to deal with the uncertainty of the information. Fuzzy logic allows one to model uncertainty and subject concepts in a better form than crisp models.

In the field of image processing, especially in industrial applications, we deal with subjective concepts like brightness, edges, uniformity, measurements, etc. Therefore, a vision inspection system used in this type of application needs to manage subjectivity and uncertainty. For example, the definition of border pixels is a task with ambiguity.

The decision whether a pixel belongs to the background or to the object is non-trivial on occasion due to the uncertainty present on object edges. The results of “crisp” algorithms in image analysis tasks are unsatisfactory if robust systems are required.

The development of new algorithms that involve fuzzy logic are required. This is why incorporation of fuzzy logic into the development of image processing and analysis has opened a new research area in the image-processing field.

* **Data Mining**

Data mining is a process of nontrivial extraction of previously unknown, implicit and potentially useful information such as knowledge rules, constraints, and regularities from data. Data mining techniques are being implemented widely on existing software and hardware platforms to enhance the value of existing information resources, and can be integrated with new products and systems as they are brought on-line.

When implemented on high performance client/server or parallel processing computers, data mining tools can analyze massive databases to deliver answers to questions. The search is performed in databases or data warehouses. The purpose is to find homogeneous categories, prototypical behaviors, and important features for the recognition of a class of data.

Using fuzzy sets brings flexibility in knowledge representation, interpretability in the obtained results, in rules or in characterizations of prototypes. Looking for too strict a relation between variables may be impossible. This is because of the variability of descriptions in the database.

While looking for an imprecise relation between variables or to a crisp relation between approximate values of variables may lead to a solution. The expressiveness of fuzzy rules or fuzzy values of attributes in a simplified natural language is a major quality for the interaction with the final user.

* **Information Retrieval**

In information retrieval, the user knows approximately what he looks for, for instance an answer to a question, or documents corresponding to a given requirement in a database. The search is performed in text, multimedia documents (images, videos, sound) or in web pages. The main difficulty lies in the identification of relevant information, i.e. the closest or the most similar to the user’s need or expectation.

The concept of relevance is very difficult to deal with, mainly because it is strongly dependent on the context of the search and the purpose of the action launched on the basis of such expected relevant information. Asking the user to elicit what he looks for is not an easy task and, the more flexible the query-answer process, the more efficient the retrieval.

This is a first reason to use fuzzy sets in knowledge representation to enable the user to express his expectations in a language not far from natural. The second reason lies in the approximate matching between the user’s queries and existing elements in the database, on the basis of similarities and degrees of satisfiability.

The main problems in information retrieval and data mining lie in the large scale of databases, especially when dealing with video or web resources, in the heterogeneous data of various types, numerical or symbolic, precise or imprecise, ambiguous, approximate, with incomplete files, uncertain because of the poor reliability of sources or the difficulties of measurement of observation.

Another source of problems is the complexity of the user’s requests, expressed in natural language or involving various criteria for instance. The necessity to create cooperative systems, friendly and user-oriented, adapted to the user’s needs or capabilities, providing a personalized information, leads to soft approaches to man-machine interaction and to on-line or off-line learning of the best way to satisfy the demand.

Fuzzy logic is useful in this matter because of its capability to represent miscellaneous data in a synthetic way, its robustness with regard to changes of the parameters of the user’s environment, and obviously its unique expressiveness.

* **Cryptography**

Today the security is the main issue in data communication. Encryption can provide a fine solution for it. The encryption algorithm is the mathematical procedure for performing encryption on data. A key is used to cipher a message and to decipher it back to the original message.

The currently used encryption algorithms only concern about security. But consuming a less processing power is also equally important as the security for connections with low bandwidths.

The algorithm implemented using fuzzy logic supports for user desired security level and processing level. It is a block cipher which is a derivation on the feistel network architecture. The algorithm provides security levels and their corresponding processing levels by using various keys for the encryption/decryption process.

The encryption algorithm using fuzzy logic provides either low processing or high security according to user’s requirement which is more advanced than the existing encryption algorithms.  Fuzzy logic provides a simple way to arrive at a definite conclusion based upon vague, ambiguous, imprecise, noisy, or missing input information.

To achieve security and low processing, the algorithm uses variable keys. 0th position gives a fully low processing algorithm and 1st position gives fully secured algorithm. The fuzzification changes depending on the key size and the number of mapping tables of the encryption algorithm.

* **Machine Learning**

Fuzzy logic is primarily used for the purpose of knowledge representation, and inference is mostly of a deductive nature. Machine learning, on the other hand, is mainly concerned with inductive inference, namely, the induction of general, idealized models from specific, empirical data.

Thus, the role of fuzzy logic in this field is much less obvious at first sight. the connection between the fuzzy and the core machine learning community is not well established at all. Related to the lack of communication between the communities, the recognition of fuzzy logic inside machine learning is still rather moderate.

The bulk of contributions in fuzzy machine learning deals with the fuzzy extension of standard, non-fuzzy methods: from rule induction to fuzzy rule induction. In general, this means an extension of the representation of corresponding models by means of fuzzy concepts, such as the use of fuzzy instead of crisp partitions in decision tree learning.

The effect is an increased flexibility of the model class, which can indeed be useful and improve performance, especially if the original class is quite restricted. Interpretability is one of the core arguments often put forward by fuzzy scholars in favor of fuzzy models—usually in a very uncritical way.

In fact, many authors seem to take it for granted that fuzzy models or, more specifically, fuzzy rule-based models, can easily be understood and interpreted by a human user or data analyst. Fuzzy sets are connected to uncertainty modeling via possibility theory, that is, via the interpretation of membership functions as possibility distributions. More generally, alternative uncertainty formalisms, typically based on non-additive measures, have been studied extensively in the fuzzy community.

Since uncertainty is inherent in inductive inference and, therefore, learning from data is inseparably connected with uncertainty, this is another opportunity to contribute to machine learning. Instead of assuming fuzzy data to be given right away, precise data is systematically “fuzzified” so as to modulate the influence of individual observations on the process of model induction.

* **Computer Security and Forensics**

Due to the growing reliance that corporations and government agencies place on their computer networks, the significance of defending these systems from attack cannot be underestimated. A single malicious encroachment into a computer network can cause a great deal of damage to an organization.

Computer Security concerns the protection of information and property from misuse by unauthorized people. Computer forensics provides methods for the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the reconstruction of criminal activities or policy violation.

Fuzzy logic can properly be used to help deal with issues associated with computer security and computer forensics. Fuzzy Logic can help us provide some good security analysis dues to its ability to take advantage of an operator’s experience.Another aspect of fuzzy Logic is the speed of response of the system, which is the key factor in computer security.

Fuzzy logic can be used to predefine attacks and to store them. These stored attacks can in turn be used when a new attack appears, by comparing it with previously stored ones. Fuzzy Logic is used in Computers to recognize assaults and compare them with attack signatures which have been stored in databases helping to determine whether security has been compromised.

**Fuzzy Logic for Image Processing**

**Introduction**

Image processing algorithms require modeling of complex systems, which require processing of information with high degree of uncertainty and subjectivity. Some algorithms developed under the scheme of traditional logic have crisp approach. As a consequence, information uncertainty and subjectivity are difficult to model in the algorithms.

A better structure to model the human visual system must incorporate tools to model better subjective concepts and must be able to deal with the uncertainty of the information. A mathematical framework that incorporates these tools is fuzzy logic.  The concepts related to image analysis contain a certain amount of uncertainty.

For example, that the definition of border pixels is a hard task. The decision whether a pixel belongs to the background or to the object is uncertain. If robust systems are required, development of new algorithms that involve fuzzy logic are required. This is why incorporation of fuzzy logic into the development of image processing and analysis has opened a new research area in the image-processing field.

Fuzzy image processing consists of Image fuzzification, Modification of membership values, and if necessary, Image defuzzification. The fuzzification and defuzzification steps are due to the fact that we do not possess fuzzy hardware. Therefore, the coding of image data (fuzzification) and decoding of the results (defuzzification) are steps that make possible to process images with fuzzy techniques.

The main power of fuzzy image processing is in the modification of membership values. After the image data are transformed from gray level plane to the membership plane (fuzzification), appropriate fuzzy techniques modify the membership values. This can be a fuzzy clustering, a fuzzy rule-based approach, a fuzzy integration approach and so on.

**Image Edge Detection**

Segmentation is the process that subdivides an image into a number of uniformly homogeneous regions. If edges of the image can be extracted and linked, the region is described by the edge contour that contains it.

Most of the traditional edge-detection algorithms in image processing typically convolute a filter operator and the input image, and then map overlapping input image regions to output signals which lead to considerable loss in edge detection; however there is no such loss in the fuzzy based method described here.

The method described here uses a fuzzy based logic model with the help of which high performance is achieved along with simplicity in resulting model. The fuzzy logic approach for image processing allows us to use membership functions to define the degree to which a pixel belongs to an edge or a uniform region.

**Steps for Edge Detection**

**Step 1: Import RGB Image and Convert to Grayscale**

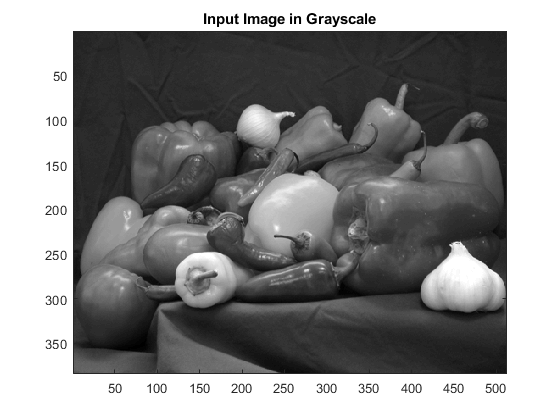
Import the image into MATLAB.

*Icolor = imread('peppers.png')*

Icolor is a 384 x 512 x 3 uint8 array. The three channels of Icolor (third array dimension) represent the red, green, and blue intensities of the image.

Convert Icolor to grayscale so that we can work with a 2-D array instead of a 3-D array. The standard NTSC conversion formula is used to calculate the effective luminance of each pixel.

*Igray1 = 0.2989\*Icolor(:,:,1)+0.5870\*Icolor(:,:,2)+0.1140\*Icolor(:,:,3);  
 figure  
 image(Igray1,'CDataMapping','scaled');  
 colormap('gray')  
 title('Input Image Grayscale')*



**Step 2: Convert Image to Double-Precision Data**

The Fuzzy Logic Toolbox software operates on double-precision numbers only. So, we convert Igray1, a uint8 array, to a double array.

*I = double(Igray1);*

Because uint8 values are in the [0 2^8-1] range, all elements of I are in that range too. Scale I so that its elements are in the [0 1] range.

*clastyp = class(Igray1);  
sf = double(intmax(clastyp));  
I = I/sf;*

**Step 3: Obtain Image Gradient**

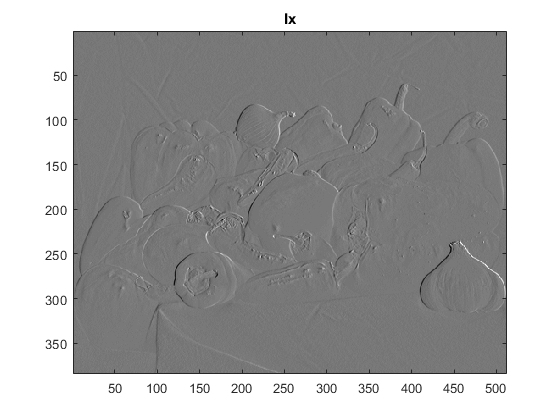
The fuzzy logic edge-detection algorithm for this example relies on the image gradient to locate breaks in uniform regions. Calculate the image gradient along the *x*-axis and *y*-axis.

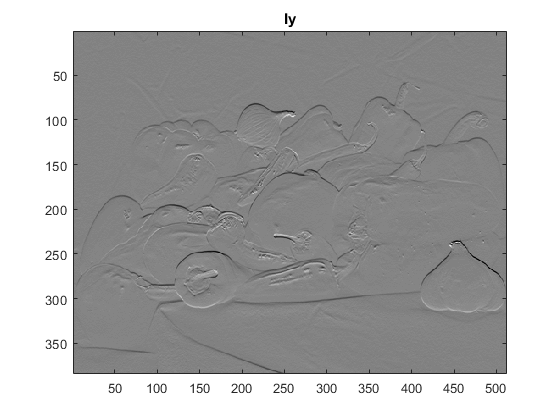
*Gx = [-1 1];  
Gy = Gx';  
Ix = conv2(I,Gx,'same');  
Iy = conv2(I,Gy,'same');*

*figure  
image(Ix,'CDataMapping','scaled')  
colormap('gray')  
title('Ix')*

*figure  
image(Iy,'CDataMapping','scaled')  
colormap('gray')*

*title('Iy')*





Filtering is a technique for modifying or enhancing an image. Filtering is a *neighborhood operation,* in which the value of any given pixel in the output image is determined by applying some algorithm to the values of the pixels in the neighborhood of the corresponding input pixel.

A pixel's neighborhood is some set of pixels, defined by their locations relative to that pixel. Linear filtering of an image is accomplished through an operation called *convolution*. Convolution is a neighborhood operation in which each output pixel is the weighted sum of neighboring input pixels. Gx and Gy are simple gradient filters.

We convolve I with Gx, using the conv2 function, to obtain a matrix containing the *x*-axis gradients of I. The gradient values are in the [-1 1] range. Similarly, we convolve I with Gy to obtain the *y*-axis gradients of I.

**Step 4: Define Fuzzy Inference System (FIS) for Edge Detection**

Create a fuzzy inference system (FIS) for edge detection, edgfis.

*edgfis = newfis('edgeDetection');*

Specify the image gradients, Ix and Iy, as the inputs of edgfis.

*edgfis = addvar(edgfis,'input','Ix',[-1 1]);  
 edgfis = addvar(edgfis,'input','Iy',[-1 1]);*

Specify a zero-mean Gaussian membership function for each input. If the gradient value for a pixel is 0, then it belongs to the zero membership function with a degree of 1.

*sx = 0.1;  
 sy = 0.1;  
 edgfis = addmf(edgfis,'input',1,'zero','gaussmf',[sx 0]);  
 edgfis = addmf(edgfis,'input',2,'zero','gaussmf',[sy 0]);*

sx and sy specify the standard deviation for the zero membership function for the Ix and Iy inputs. We can change the values of sx and sy to adjust the edge detector performance. Increasing the values makes the algorithm less sensitive to the edges in the image and decreases the intensity of the detected edges. We specify the intensity of the edge-detected image as an output of edgfis.

*edgfis = addvar(edgfis,'output','Iout',[0 1]);*

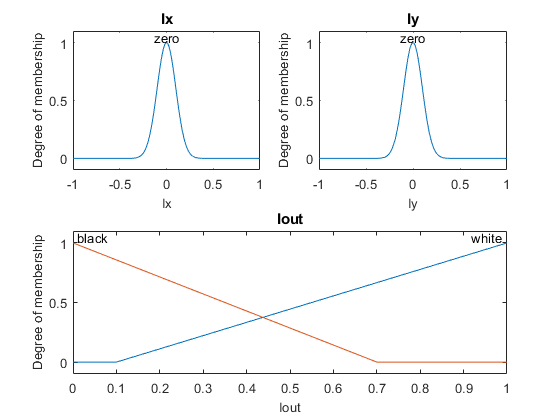
Specify the triangular membership functions, white and black, for Iout.

*wa = 0.1;  
wb = 1;  
wc = 1;  
ba = 0;  
bb = 0;  
bc = 0.7;  
edgfis = addmf(edgfis,'output',1,'white','trimf',[wawbwc]);  
edgfis = addmf(edgfis,'output',1,'black','trimf',[ba bb bc]);*

We can change the values of wa, wb, wc, ba, bb, and bc to adjust the edge detector performance. The triplets specify the start, peak, and end of the triangles of the membership functions. These parameters influence the intensity of the detected edges.

Plot the membership functions of the inputs/outputs of edgfis.

*figure  
subplot(2,2,1)  
plotmf(edgfis,'input',1)  
title('Ix')  
subplot(2,2,2)  
plotmf(edgfis,'input',2)  
title('Iy')  
subplot(2,2,[3 4])  
plotmf(edgfis,'output',1)  
title('Iout')*



**Step 5: Specify FIS Rules**

We add rules to make a pixel white if it belongs to a uniform region. Otherwise, make the pixel black.

*r1 = 'If Ix is zero and Iy is zero then Iout is white';  
r2 = 'If Ix is not zero or Iy is not zero then Iout is black';  
r = char(r1,r2);  
edgfis = parsrule(edgfis,r);  
showrule(edgfis)*

ans =  2x67 char array

'1. If (Ix is zero) and (Iy is zero) then (Iout is white) (1)       '  
 '2. If (Ix is not zero) or (Iy is not zero) then (Iout is black) (1)'

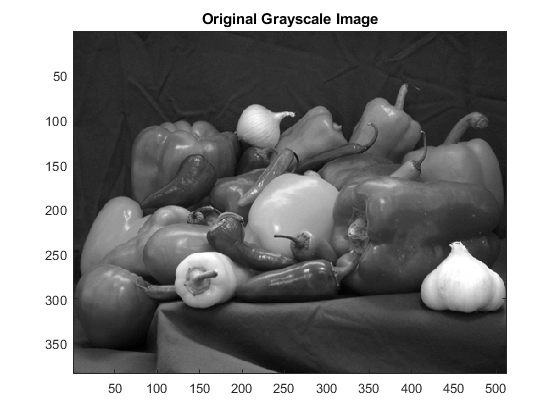
**Step 6: Evaluate FIS**

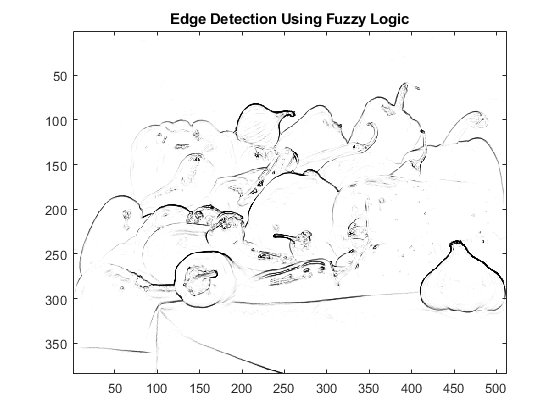
Evaluate the output of the edge detector for each row of pixels in I using corresponding rows of Ix and Iy as inputs.

*Ieval = zeros(size(I));  
 for ii = 1:size(I,1)  
   Ieval(ii,:) = evalfis([(Ix(ii,:));(Iy(ii,:));]',edgfis);  
 End*

**Step 7: Plot Results**

*figure  
image(I,'CDataMapping','scaled')  
colormap('gray')  
title('Original Grayscale Image')  
  
figure  
image(Ieval,'CDataMapping','scaled')  
colormap('gray')  
title('Edge Detection Using Fuzzy Logic')*





**Fuzzy Logic in Computer Security and Forensics**

**Introduction**

Due to the growing reliance that corporations and government agencies place on their computer networks, the significance of defending these systems from attack cannot be underestimated. A single malicious encroachment into a computer network can cause a great deal of damage to an organization.

Computer Security concerns the protection of information and property from misuse by unauthorized people. Computer forensics provides methods for the preservation, collection, validation, identification, analysis, interpretation, documentation and presentation of digital evidence derived from digital sources for the reconstruction of criminal activities or policy violation.

Fuzzy logic can properly be used to help deal with issues associated with computer security and computer forensics. Computer Security concerns the protection of information and property from misuse by unauthorized people.

Fuzzy Logic can help us provide some good security analysis dues to its ability to take advantage of an operator’s experience.Another aspect of fuzzy Logic is the speed of response of the system., which is the key factor in computer security.

Fuzzy logic can be used to predefine attacks and to store them. These stored attacks can in turn be used when a new attack appears, by comparing it with previously stored ones. Fuzzy Logic is used in Computers to recognize assaults and compare them with attack signatures which have been stored in databases helping to determine whether security has been compromised.

Security threats can be detected by using combination of Fuzzy Inference Rules and linguistic variables without the need to use clear mathematical formula which often takes great work to develop. For example, a set of rules can be taken as set of conditions on which the result depends, such as:

If an input is unknown and behavior is unfamiliar Then the risk is very likely.

Because they account for partial degrees of membership, fuzzy systems can classify inputs depending on the risk to our system. Most systems that have security function will discover whether the input is normal or not normal.

However Fuzzy Logic can be used to take this step further and it can classify abnormal inputs as being low, medium or very dangerous attacks. This can be achieved by using Fuzzy Inference Rules to the Security systems along with other rules. This way, attacks can be classified according to degree of dangerousness.

These fuzzy rules can be written by security administrators using their experience.Fuzzy Expert then then the rules cover all inputs.

Therefore, attacks can easily be detected with fuzzy systems based on the set rules defined from the expert knowledge of security administrator.

Computer viruses and malicious software can be detected by traditional anti-virus software. The problem with these programs is that they can seek only specific virus patterns; in other words, unknown computer viruses cannot be detected putting the system at risk. This has opened new axis of research including behavior - based systems. These systems can close the gap between behavior-based systems and knowledge based systems.

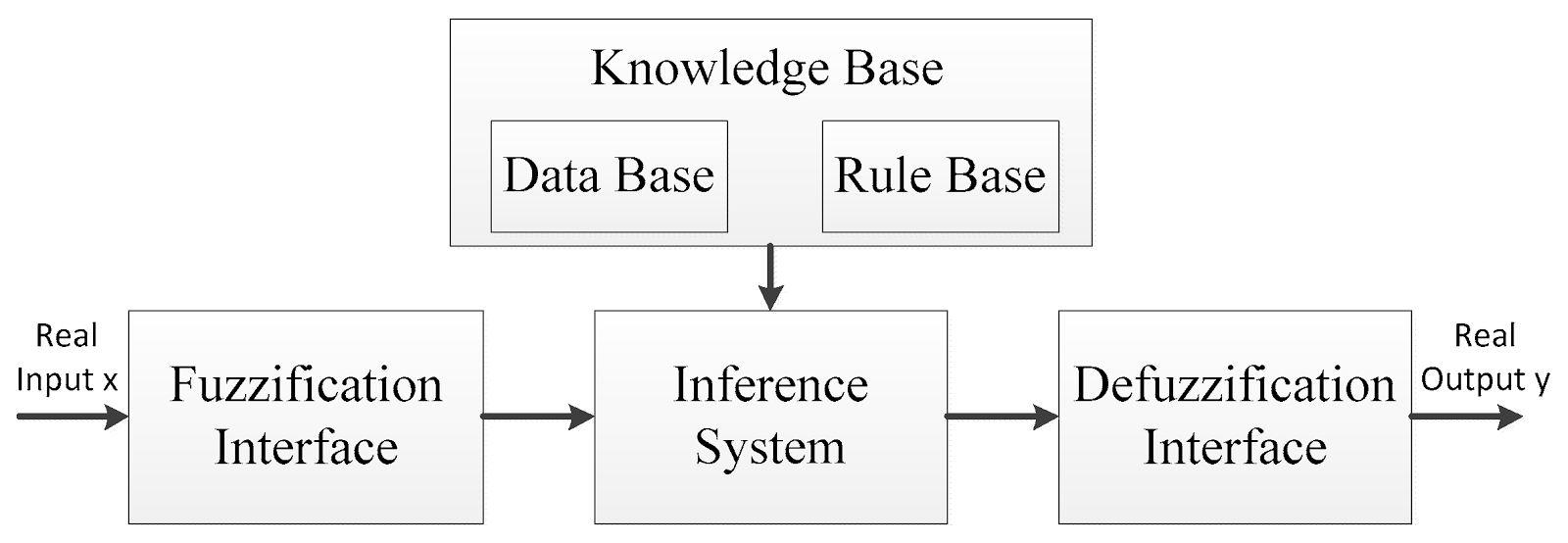
Fuzzy Logic Systems such as fuzzy pattern recognition can be used to detect unknown malicious software and computer viruses based on their behavior.

**Fuzzy Logic Control**

Fuzzy Inference system refers to the process leading to evaluation of fuzzy rules of the form:

If x is A AND y is B is given,then z is C.

Where A,B and C are fuzzy sets.



A fuzzy rule system,known as fuzzy logic control (FLC) is the one to decide over many complex problems.

The basic idea behind FLC is to introduce “expert experience “of human operator in defining the rule and inferring the relation between inputs and outputs. It consists of following:

*Knowledge Base*

It consists of definition of the universe of discourse for each variable, the number of fuzzy set representations of input-output variables and the design of the membership functions to use.

*Fuzzification Unit*

It transforms the crisp measured data (e.g. velocity is 167 km/hr) to suitable fuzzy sets (i.e speed is too fast).

*Rule Base*

It stores the observed knowledge of the operation of the process which contains a set of ‘if - then ‘rules.

*Inference Engine*

It is the main heart of fuzzy logic control and is capable of simulating human decision making by performing approximate reasoning.

*Defuzzification Unit*

Defuzzification unit covers the Fuzzy Control action to a non - fuzzy action that can be accepted by real world.

**Intrusion Detection**

Fuzzy systems have a number of aspects which make them suitable for intrusion detection. Listed are some:

1. They have ability to combine inputs taken from different sources.
2. They can cope with intrusions where some encroachments cannot be clearly determined.
3. Fuzzy systems have the ability to recognize the level of the attack because it deal with membership grade functions.

In addition, fuzzy rule detection system provides a perfect method to capture the imprecise tendency of occurrences in the computer network system.

Fuzzy Logic is well suited for situations when the differences between normal and irregular classes are not specified. This happens when Fuzzy spaces enable an object to be member of different classes at the same time.

Intrusion detection systems need to take in account the number of quantitative characteristics. So, in order to recognize high-level patterns, fuzzy set theory needs the ability to classify these quantitative characteristics. In addition, in quantitative characteristics, normal and abnormal cannot be directly separated. As a result, fuzzy systems can be used to distinguish between quantitative and non - quantitative characteristics based on fuzzy rules.

Anomalous behavior can be detected by means of statistical analysis. However, normal patterns of statistical features can also be constructed automatically and more efficiently using fuzzy systems. These systems are then used for anomaly detection. The Fuzzy Intrusion Recognition Engine (FIRE) is an example of an intrusion detection system which is used to detect anomalous behavior using fuzzy principles to recognize malevolent network action.

FIRE uses autonomous fuzzy agents which work together to supervise the incoming data in the network. Depending on past abnormal behavior which has been detected, an agent can detect whether these inputs are normal or not.

Fuzzy Logic is human like reasoning approach that resembles the ambiguity of human daily life. It has the ability to describe aspects that may not be apparent i.e., it can provide precise outputs by its ability to assess degrees of membership and linguistic variables. Security in itself is imprecise, and therefore fuzzy system provide a sound approach towards solving the problems which are related to computer security and computer forensics.

There are two types of intrusion detection: misuse detection and anomaly detection (Sundaram 1996). Misuse detection can be applied to the attacks that generally follow some fixed patterns. For example, three consecutive login failures are likely to be one of the important characteristics of password guessing.

Misuse detection is usually constructed to examine these intrusion patterns that have been recognized and reported by experts. However, intruders do not always follow publicly known patterns to break into a computer system. They will often try to mask their illegal behavior to deceive the detection system. Anomaly detection methods are designed to counter this kind of challenge.

Unlike misuse detection that is based on attack patterns, anomaly detection tries to find patterns of normal behavior, with the assumption that an intrusion will usually include some deviation from this normal behavior. Observation of this deviation will then result in an intrusion alarm.

Artificial intelligence (AI) techniques have played an important role in both misuse detection and anomaly detection. AI techniques can be used for data reduction and classification tasks (Frank 1994). For example, many intrusion detection systems have been developed as rule-based expert systems.

An example is SRI’s Intrusion Detection Expert System (IDES) (Lunt and Jagannathan 1988). The rules for detection can be constructed based on the knowledge of system vulnerabilities or known attack patterns.

On the other hand, AI techniques also have the capability of learning inductive rules. For example, sequential patterns can be learned by a system such as the Time- 3 based Inductive Machine (TIM) for intrusion detection (Teng, Chen, and Lu 1990). Neural networks can be used to predict future intrusions after training (Debar, Becker, and Siboni 1992). Data mining methods, such as association rules and frequency episodes, have been also proposed to mine normal patterns from audit data (Lee, Stolfo, and Mok 1998)

Fuzzy Logic is a human - like reasoning approach that resembles the ambiguity of human daily life. It has the ability to describe core aspects that may not be apparent ie.it can be used to provide precise outcomes by its ability to access degree of membership and the inference rules. Security in itself is imprecise, and therefore fuzzy logic provide a sound approach for solving problems related to fuzzy logic.

**FUZZY RETRIEVAL**

In information retrieval, the user knows approximately what he looks for, for instance an answer to a question, or documents corresponding to a given requirement in a database. The search is performed in text, multimedia documents (images, videos, sound) or in web pages. The main difficulty lies in the identification of relevant information, i.e. the closest or the most similar to the user’s need or expectation.

The concept of relevance is very difficult to deal with, mainly because it is strongly dependent on the context of the search and the purpose of the action launched on the basis of such expected relevant information. Asking the user to elicit what he looks for is not an easy task and, the more flexible the query-answer process, the more efficient the retrieval.

This is a first reason to use fuzzy sets in knowledge representation to enable the user to express his expectations in a language not far from natural. The second reason lies in the approximate matching between the user’s queries and existing elements in the database, on the basis of similarities and degrees of satisfiability.

The main problems in information retrieval and data mining lie in the large scale of databases, especially when dealing with video or web resources, in the heterogeneous data of various types, numerical or symbolic, precise or imprecise, ambiguous, approximate, with incomplete files, uncertain because of the poor reliability of sources or the difficulties of measurement of observation.

Another source of problems is the complexity of the user’s requests, expressed in natural language or involving various criteria for instance. The necessity to create cooperative systems, friendly and user-oriented, adapted to the user’s needs or capabilities, providing a personalized information, leads to soft approaches to man-machine interaction and to on-line or off-line learning of the best way to satisfy the demand.

Fuzzy logic is useful in this matter because of its capability to represent miscellaneous data in a synthetic way, its robustness with regard to changes of the parameters of the user’s environment, and obviously its unique expressiveness.

**Fuzzy retrieval** techniques are based on the Extended Boolean Model and the Fuzzy Set theory. There are two classical fuzzy retrieval models:

1. Mixed Min and Max (MMM) and
2. Paice model.

Both models do not provide a way of evaluating query weights, however this is considered by the P-norms algorithm.

MMM

In fuzzy-set theory, an element has a varying degree of membership, say  to a given set *A* instead of the traditional membership choice (is an element/is not an element).  
In MMM each index term has a fuzzy set associated with it. A document's weight with respect to an index term *A* is considered to be the degree of membership of the document in the fuzzy set associated with *A*. The degree of membership for union and intersection are defined as follows in Fuzzy set theory:{\displaystyle d\_{A\cap B}=min(d\_{A},d\_{B})}

{\displaystyle d\_{A\cup B}=max(d\_{A},d\_{B})}According to this, documents that should be retrieved for a query of the form *A or B*, should be in the fuzzy set associated with the union of the two sets *A* and *B*. Similarly, the documents that should be retrieved for a query of the form *A and B*, should be in the fuzzy set associated with the intersection of the two sets. Hence, it is possible to define the similarity of a document to the *or* query to be  and the similarity of the document to the *and* query to be  . The MMM model tries to soften the Boolean operators by considering the query-document similarity to be a linear combination of the *min* and *max* document weights.

Given a document *D* with index-term weights  for terms and the queries:

the query-document similarity in the MMM model is computed as follows:

where *Cor1, Cor2* are "softness" coefficients for the *or* operator, and *Cand1, Cand2* are softness coefficients for the *and* operator. Since we would like to give the maximum of the document weights more importance while considering an *or* query and the minimum more importance while considering an *and* query, generally we have  *and* . For simplicity it is generally assumed that  and . Lee and Fox experiments indicate that the best performance usually occurs with *Cand1* in the range and with . In general, the computational cost of MMM is low, and retrieval effectiveness is much better than with the [Standard Boolean model](https://en.wikipedia.org/wiki/Standard_Boolean_model).

**PAICE MODEL**

The Paice model is a general extension to the MMM model. In comparison to the MMM model that considers only the minimum and maximum weights for the index terms; the Paice model incorporates all of the term weights when calculating the similarity:

{\displaystyle S(D,Q)=\sum \_{i=1}^{n}{\frac {r^{i-1}\*w\_{di}}{\sum \_{j=1}^{n}r^{j-1}}}}where *r* is a constant coefficient and  is arranged in ascending order for *and* queries and descending order for *or* queries. When the Paice model shows the same behavior as the MMM model.

The experiments of Lee and Fox have shown that setting the  for *and* queries and 0.7 for *or* queries gives good retrieval effectiveness. The computational cost for this model is higher than that for the MMM model. This is because the MMM model only requires the determination of *min* or *max* of a set of term weights each time an *and* or *or* clause is considered, which can be done in. The Paice model requires the term weights to be sorted in ascending or descending order, depending on whether an *and* clause or an *or* clause is being considered. This requires at least an  sorting algorithm. A good deal of floating point calculation is needed too.

**References**

1. Edge Detection: <https://link.springer.com/chapter/10.1007/978-1-84628-469-4_7>
2. Cryptography: <https://pdfs.semanticscholar.org/7500/c09ea9fd262cccf0b74eef74e719d0c085f2.pdf>
3. Cryptography:

<http://www.diva-portal.se/smash/get/diva2:725347/FULLTEXT01.pdf>

1. ShashankMathur, Anil Ahlawat. “APPLICATION OF FUZZY LOGIC ON IMAGE EDGE DETECTION”, INFOS 2008, Varna, Bulgaria, June-July 2008

5. <https://en.wikipedia.org/wiki/Fuzzy_retrieval>

**INDEX**

1. Fuzzification
   1. Fuzzy Linguistic Variables
   2. Membership functions
2. Types of Membership Functions
   1. Triangular Membership Function
   2. Trapezoidal Membership Function
   3. Gaussian And Generalized Bell Function
   4. Sigmoidal Function
   5. Polynomial Function
   6. L-R Type Membership Functions
3. Defuzzification
4. Methods of defuzzification
   1. Centre of Sums
   2. Center of Largest Area
   3. Centroid Method
   4. Weighted Average Method
   5. Max Membership Principle
   6. First (Or Last) Of Maxima

**FUZZIFICATION:**

It is the process of changing a real scalar value into a fuzzy value. Fuzzy membership functions represent likenesses of objects to vague properties. All the information represented by a fuzzy set is contained within the membership function. It is the process which allows the continuous values of variables to be changed into language variables.

This is achieved with different types of*fuzzifiers (membership functions).*

**Fuzzy Linguistic Variables**

They are used to represent qualities spanning a particular spectrum.

Temp: {Freezing, Cool, Warm, Hot}

*here, temp is a linguistic variable and* {freezing, cool, warm, hot} are liguistic values.

Illustration:



**Membership functions:**

The membership function is used to associate a grade to each linguistic term. It lies between .

let’s take an example,

eg-1.For the fuzzification of the car speed value .two membership functions can be used, which characterize a low and a medium speed fuzzy set, respectively. The given speed value of belongs with a grade of to the fuzzy set ``low'' and with a grade of to the fuzzy set ``medium".

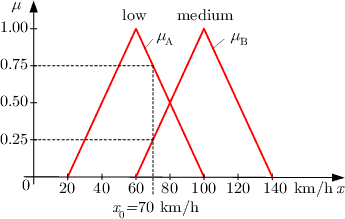


fig 1.fuzzification of a car speed

**Types of Membership Functions**

1. Triangular function
2. Trapezoidal function
3. Gaussian function
4. Generalized bell function
5. Sigmoidal function
6. Polynomial function

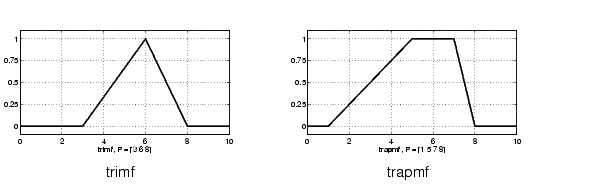
* S-function
* Z-function
* Π-function

1. L-R type function

**Triangular Membership Function**:

The simplest membership functions are formed using straight lines. The simplest membership function is the triangular membership function, and it has the function name trimf. It's nothing more than a collection of three points forming a triangle.

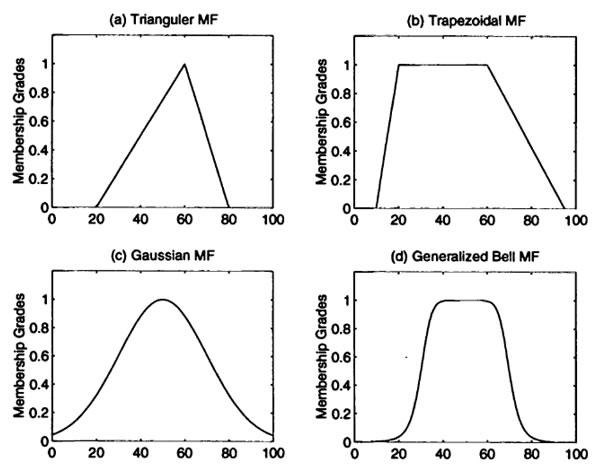
Representation of triangular membership function with help of graph:



A triangular membership function is specified by three parameters (a, b, c) as follows:

The parametexrs determine the x coordinates.

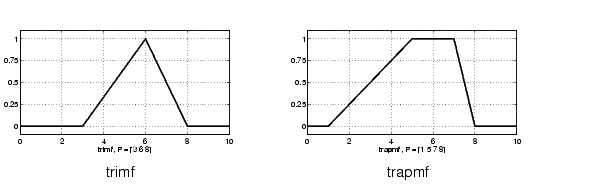
* Illustration of triangular MF with help of example:



**Trapezoidal Membership Function:**

The trapezoidal membership function has a flat top and really is just a truncated triangle curve. It has function name trapmf. These straight line membership functions have the advantage of simplicity.

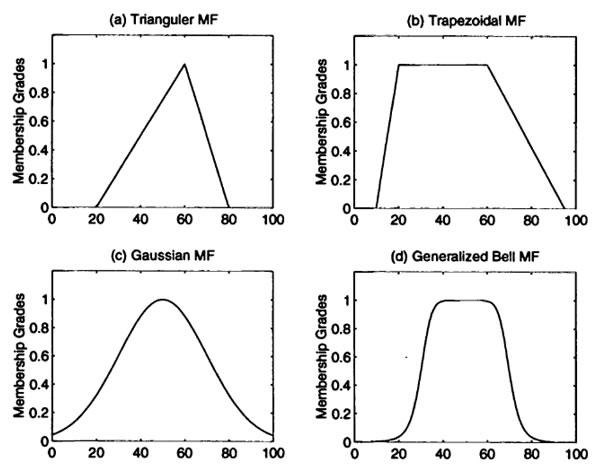
Representation of trapezoidal membership function with help of graph:



A trapezoidal membership function is specified by four parameters (a, b, c, d) as follows:

The parameters determine the x coordinates of the four corners of the underlying trapezoidal MF.

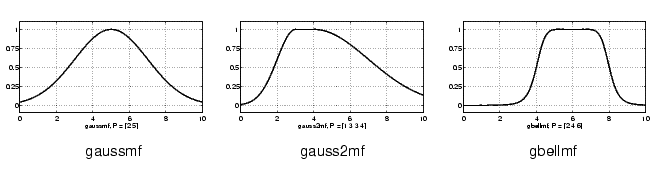
* Illustration of trapezoidal membership function by example:



**Gaussian And Generalized Bell Function:**

* Two membership functions are built on the Gaussian distribution curve: a simple Gaussian curve and a two-sided composite of two different Gaussian curves. The two functions are gaussmf and gauss2mf. It is specified by two parameters. Gaussian functions are well known in probability and statistics. They possess useful properties such as invariance under multiplication (the product of two Gaussians is a Gaussian with a scaling factor) and Fourier transform (the Fourier transform of a Gaussian is still a Gaussian).
* The generalized bell membership function is specified by three parameters. It has the function name gbellmf. The bell membership function has one more parameter than the Gaussian membership function, so that it can approach a non-fuzzy set if the free parameter is tuned. Gaussian and bell membership functions are popular methods for specifying fuzzy sets because of their smoothness and concise notation. Both of these curves have the advantage of being smooth and nonzero at all points.They are unable to specify asymmetric membership functions.

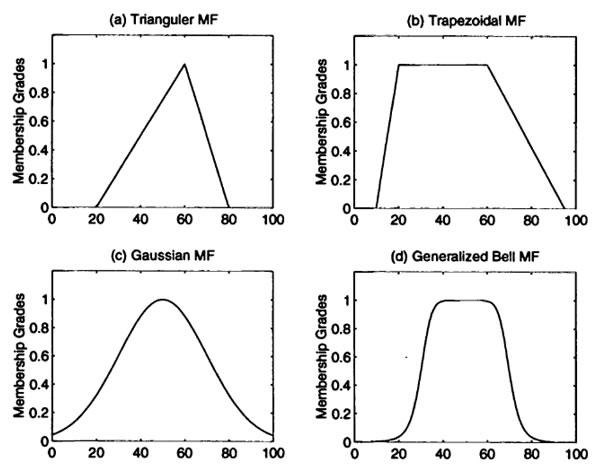
Representation of Gaussian and generalized bell membership function with help of graph:



A Gaussian membership function is specified by two parameters:

A Gaussian MF is determined complete by and; represents the MFs centre and σ determines the MFs width.

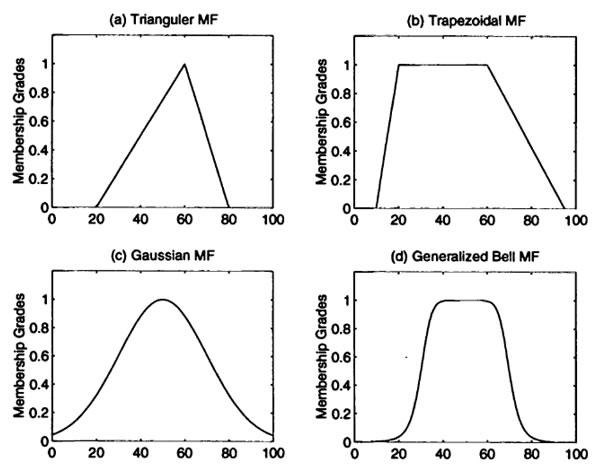
* Illustration of Gaussian membership function with example:



A generalized bell MF (or [Bell-shaped Function](http://researchhubs.com/post/maths/fundamentals/bell-shaped-function.html)) is specified by three parameters {a, b, c}:

where parameter b is usually positive. (If b is negative, then the shape of this MF becomes an upside-down bell.)

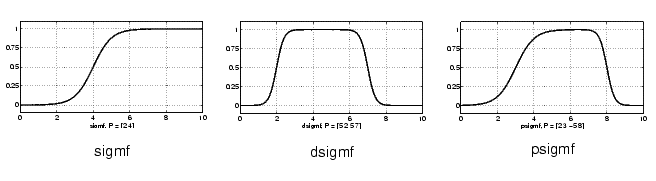
* Illustration of bell shaped membership function with help of example:



**Sigmoidal Function:**

 It is either open left or right. Asymmetric and closed (i.e. not open to the left or right) membership functions can be synthesized using two sigmoidal functions, so in addition to the basic sigmf, we also have the difference between two sigmoidal functions, dsigmf, and the product of two sigmoidal functions psigmf.

Representation of Sigmoidal membership function with help of graph:



A sigmoidal function is defined by two parameters (a, c) as follows:

where a controls the slop at the crossover point x=c

**Polynomial Function:**

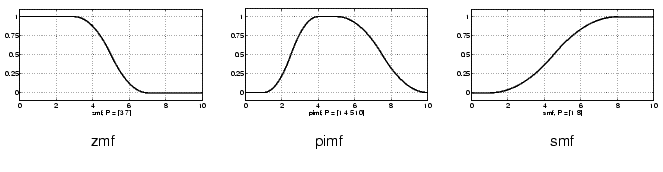
They account for several of the membership functions.

Three related membership functions are the:

1. Z-Shaped
2. S-Shaped
3. Π-Shaped

All three are named because of their shapes. The function zmf (z-shaped) is the asymmetrical polynomial curve open to the left,smf (s-shaped) is the mirror-image function that opens to the right, and pimf ( π-shaped) is zero on both extremes with a rise in the middle.

Representation of various polynomial membership function with help of graph:



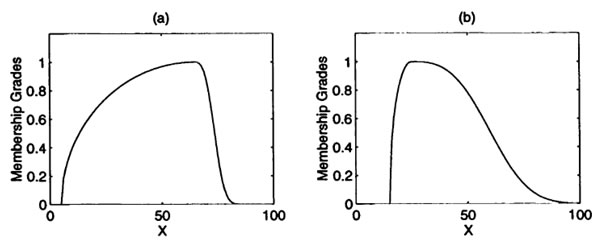
**L-R Type Membership Functions:**

A left-right MF or L-R MF is specified by three parameters {α, β, c}:

) x ≥ c

X is the modal value of the membership function and  and β are the spreads corresponding to the left-hand and right-hand curve of the membership function respectively.

Eg. and



As an abbreviated notation, we can define an L-R fuzzy number  with the membership function by

**DEFUZZIFICATION**

It is the process of producing a quantifiable result in Crisp logic, given fuzzy sets and corresponding membership degrees. It is the process that maps a fuzzy set to a crisp set. It is inverse of fuzzification.

**Methods of defuzzification**

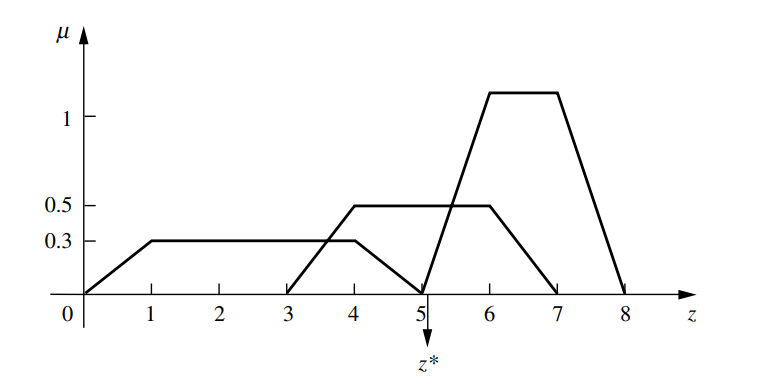
**Center of Sums**

It is the fastest defuzzification method. This process involves the algebraic sum of individual output fuzzy sets. The only drawback of this method is that intersecting areas are added twice.

The center of sums weights are the areas of the respective membership functions

The algebraic equation for defuzzified value is given by the equation:

* Illustration of center of sum with help of example:



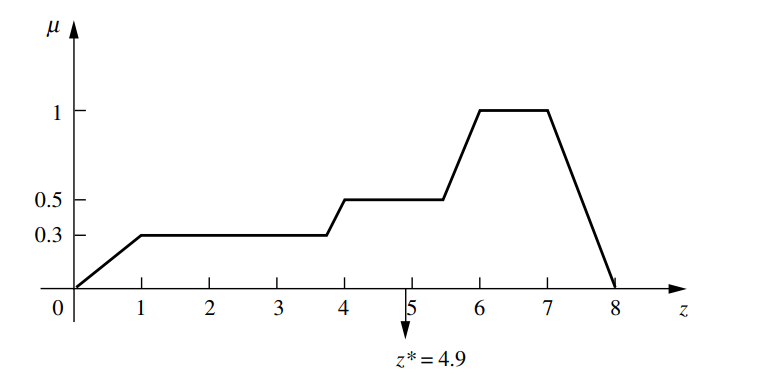
**Center of Largest Area**

If the output fuzzy set has two convex sub-regions, then the center of gravity (i.e. z\* is calculated using centroid method) of the convex fuzzy sub-region with largest area is used to obtain the defuzzified value z\* of the output.

The algebraic equation for the defuzzified value in this case is given by:

where, corresponds to that of largest area.

Illustration of above method with the help of example:

**Centroid Method**

It is the most prevalent and physically appealing of all the defuzzification methods. It was given by (Sugeno, 1985) and (Lee, 1990)

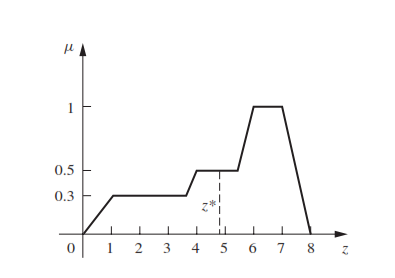
It is also known as:

1. Center of area

2. Center of gravity

The algebraic equation for the defuzzified value in this case is given by:

Illustration of above method with the help of example:



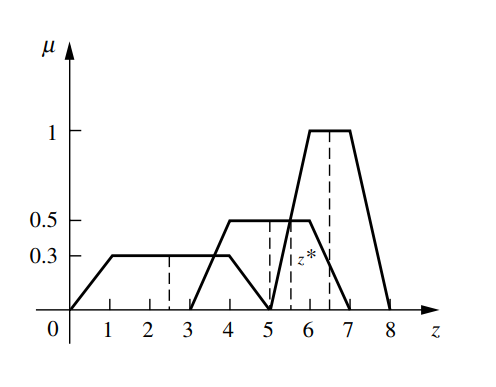
**Weighted Average Method:**

The weighted average method is the most frequently used in fuzzy applications since it is one of the more computationally efficient methods. Unfortunately, it is usually restricted to symmetrical output membership functions.

Its algebraic expression is given by:

Where ∑ denotes the algebraic sum and where ƶ is the centroid of each symmetric membership function.

Illustration of above method with help of example:



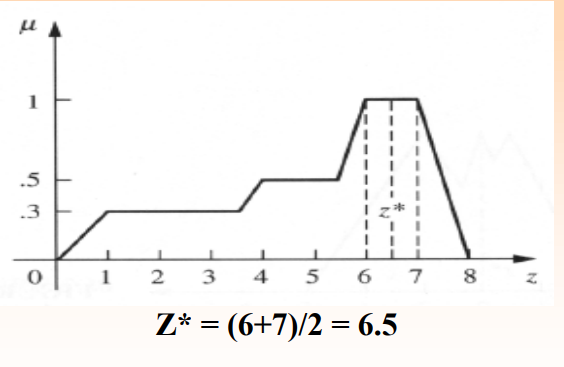
**Max Membership Principle:**

It is also known as the height method; this scheme is limited to peaked output functions.

This method is given by the algebraic expression:

where is the defuzzified value

illustration of above method with the help of example:



**Mean Max Membership**

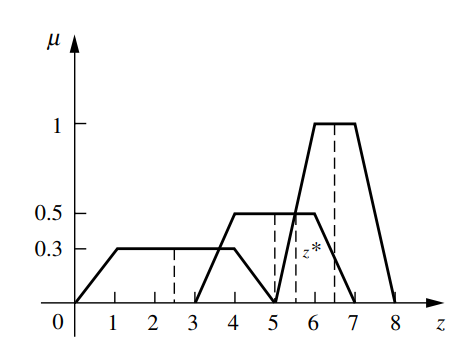
This method (also called middle-of-maxima) is closely related to the max membership principle, except that the locations of the maximum membership can be non-unique (i.e., the maximum membership can be a plateau rather than a single point).

This method is given by the expression:

Illustration of above method with help of example:

According to the mean max membership method,

is given by



**First (Or Last) Of Maxima**

This method uses the overall output or union of all individual output fuzzy sets CK to determine the smallest value of the domain with maximized membership degree in CK .

The equations for z∗ are as follows:

First, the largest height in the union (denoted hgt(CK) is determined,

Then, first of the maxima is found,

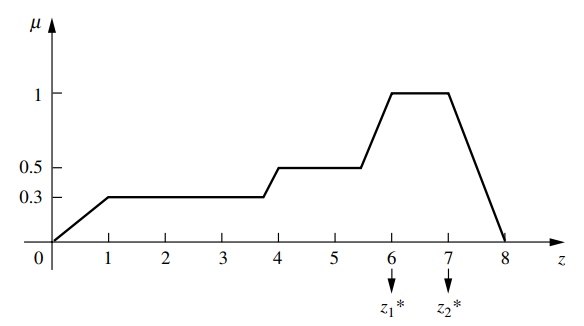
Where, the infimum (inf) is the greatest lower bound

An alternative to this method is called the last of maxima, and it is given by

Where, the supremum (sup) is the least upper bound

Illustration of above method with help of example:

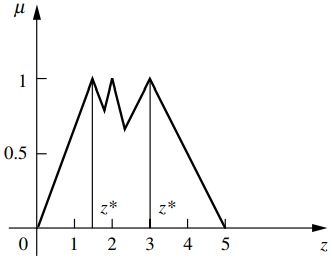
Eg-1.



First of maxima solution:

Last of maxima solution:

Eg.2

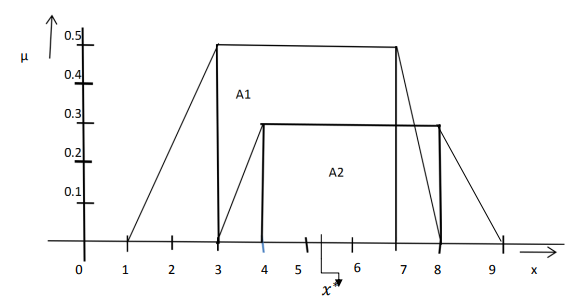


First of maxima solution:

Last of maxima solution

**MCQ:**

1. Calculate the defuzzified value of given figure using center of sum method:



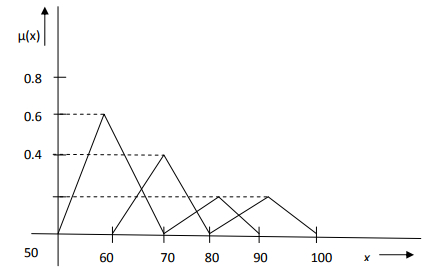
1. 5.35
2. 6.5
3. 5.75
4. 8
5. None of these

Answer: 5.35

1. For the above given figure calculate defuzzified value using centroid method.
2. 5.009
3. 5.008
4. 6.000
5. 5.004
6. None of these

Answer: 5.008

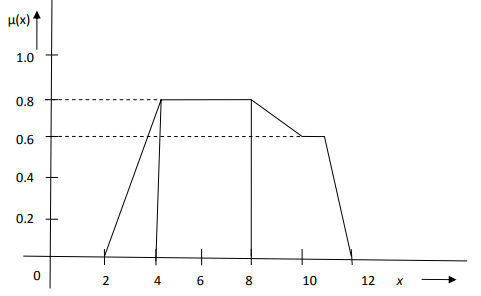
1. Calculate defuzzified value using weighted average method for given figure:



1. 75
2. 80
3. 70
4. 90
5. None of these

Answer: 70

1. Calculate the defuzzified value for given fig. using (a) first of maxima method (b) last of maxima method :



(a) For first of maxima method

1. 2
2. 4
3. 6
4. 8
5. None of these

Answer: 4

(b) For last of maxima method:

1. 4

2. 8

3. 10

4. 12

5. None of these

Answer: 8

MATLAB CODE :

Suppose you have the following region to be defuzzified. What are some of the methods you might choose?

x = -10:0.1:10;

mf1 = trapmf(x,[-10 -8 -2 2]);

mf2 = trapmf(x,[-5 -3 2 4]);

mf3 = trapmf(x,[2 3 8 9]);

mf1 = max(0.5\*mf2,max(0.9\*mf1,0.1\*mf3));

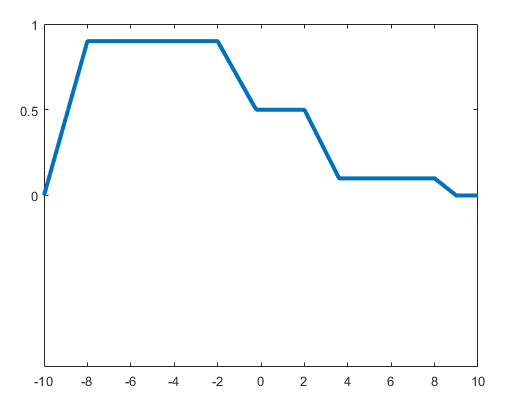
figure('Tag','defuzz');

plot(x,mf1,'LineWidth',3);

h\_gca = gca;

h\_gca.YTick = [0 .5 1] ;

ylim([-1 1]);



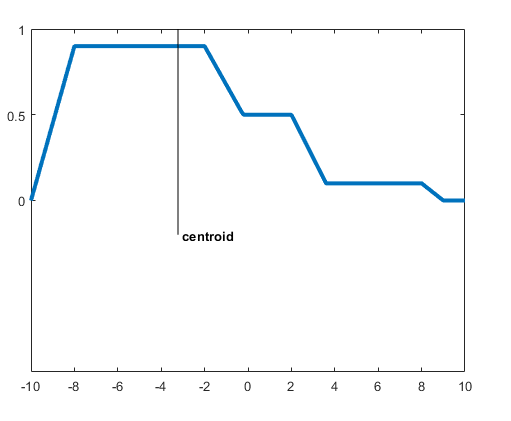
**Centroid**

Centroid defuzzification returns the center of area under the curve. If you think of the area as a plate of equal density, the centroid is the point along the x axis about which this shape would balance.

x1 = defuzz(x,mf1,'centroid'); % #ok<\*NOPTS>

h1 = line([x1 x1],[-0.2 1.2],'Color','k');

t1 = text(x1,-0.2,' centroid','FontWeight','bold');



**Middle, Smallest, and Largest of Maximum**

MOM, SOM, and LOM stand for Middle, Smallest, and Largest of Maximum, respectively. These three methods key off the maximum value assumed by the aggregate membership function. In this example, because there is a plateau at the maximum value, they are distinct. If the aggregate membership function has a unique maximum, then MOM, SOM, and LOM all take on the same value.

x3 = defuzz(x,mf1,'mom')

x3 = -5

x4 = defuzz(x,mf1,'som')

x4 = -2

x5 = defuzz(x,mf1,'lom')

x5 = -8

h2.Color = gray;

t2.Color = gray;

h3 = line([x3 x3],[-0.7 1.2],'Color','k');

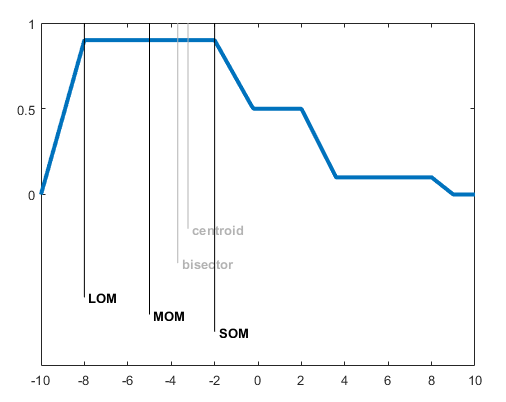
t3 = text(x3,-0.7,' MOM','FontWeight','bold');

h4 = line([x4 x4],[-0.8 1.2],'Color','k');

t4 = text(x4,-0.8,' SOM','FontWeight','bold');

h5 = line([x5 x5],[-0.6 1.2],'Color','k');

t5 = text(x5,-0.6,' LOM','FontWeight','bold');



**REFERENCES**

1. Bezdek, J. (1993). ‘‘Editorial: Fuzzy models – what are they, and why?’’ IEEE Trans. Fuzzy Syst., vol. 1, pp. 1–5.
2. Filev, D. and Yager, R. (1991). ‘‘A generalized defuzzification method under BAD distributions,’’ Int. J.Intell. Syst., vol. 6, pp. 689–697.
3. Hellendoorn, H. and Thomas, C. (1993). ‘‘Defuzzification in fuzzy controllers,’’ Intell. Fuzzy Syst., vol. 1, pp. 109–123.
4. Klir, G. and Folger, T. (1988). Fuzzy sets, uncertainty, and information, Prentice Hall, Englewood Cliffs, NJ
5. Lee, C. (1990). ‘‘Fuzzy logic in control systems: fuzzy logic controller, Parts I and II,’’ IEEE Trans. Syst., Man, Cybern., vol. 20, pp. 404–435.
6. Sugeno, M. (1985). ‘‘An introductory survey of fuzzy control,’’ Inf. Sci., vol. 36, pp. 59–83
7. N. Mogharreban and L. F. DiLalla “Comparison of Defuzzification Techniques for Analysis of Noninterval Data”, IEEE, 06.
8. Jean J. Saade and Hassan B. Diab. “Defuzzification Methods and New Techniques for Fuzzy Controllers”, Iranian Journal of Electrical and Computer Engineering, 2004.
9. AarthiChandramohan, M. V. C. Rao and M. SenthilArumugam: “Two new and useful defuzzification methods based on root mean square value”, Soft Computing, 2006.
10. Soft Computing by D.K. Pratihar, Narosa Publication.