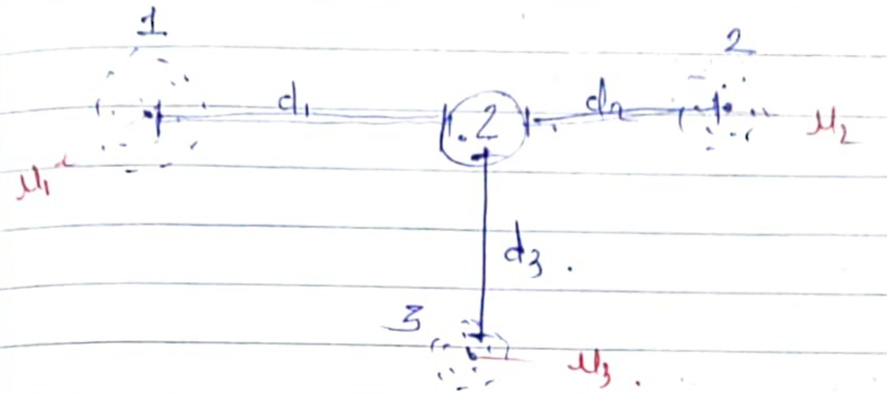
Maximum Distance Classifier (MDC).



if $d_1 < d_2$ & $d_1 < d_3$, then unknown belongs to class 1.

⟹ Given training pattern set.

1) Calculate μ_i, $i = 1, 2, \ldots M$, from the training pattern set.

2) For any x, be in class i, if

$$(x-\mu_i)^T(x-\mu_i) \leq (x-\mu_j)^T(x-\mu_j)$$
$$\forall\ i,j - 1 \text{ to } M \text{ and } i \neq j.$$

⟹ if $P_i(P_i(x)) \geq P_j(P_j(x)) \quad \forall\ i,j \in [1, M]$ & $i \neq j$.

→ Assume that,

1)
$$p_i(x) = \frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}} \exp\left\{\frac{-1}{2}(x-\mu_i)^T \Sigma^{-1}(x-\mu_i)\right\}$$
for all $i = 1$ to M.

2) $P_1 = P_2 = \ldots\ldots = P_M$

$$\Rightarrow \cancel{P_i} \cancel{\frac{1}{(2\pi)^{D/2}|\Sigma_i|^{1/2}}} \exp\left\{-\frac{1}{2}(x-\mu_i)^T\Sigma^{-1}(x-\mu_i)\right\}$$

$$> \cancel{P_j} \cancel{\frac{1}{(2\pi)^{D/2}|\Sigma_j|^{1/2}}} \exp\left\{-\frac{1}{2}(x-\mu_j)^T\Sigma^{-1}(x-\mu_j)\right\}$$

$$P_i = P_j = \frac{1}{M}.$$

$\Rightarrow$ Third Assumption,

$$\Sigma_1 = \Sigma_2 = \Sigma_3 = \cdots = \Sigma_M = I_{D\times D}$$

$\rightarrow$ Normalization.

$$\frac{x - x_{min}}{x_{max} - x_{min}}.$$

$\rightarrow$ Standardisation.

$$\boxed{y = \frac{x-\mu_x}{\sigma_x^2}.}$$

$$\Rightarrow \begin{matrix}\mu_y = 0\\ \sigma_y^2 = 1\end{matrix}$$

$$\{I^{-1} = I\}$$

$$\Rightarrow -\frac{1}{2}(x-\mu_i)^T I^{-1}(x-\mu_i) >$$

$$-\frac{1}{2}(x-\mu_j)^T I^{-1}(x-\mu_j).$$

$\Rightarrow$

Camlin

$$\Rightarrow \boxed{(x-\mu_i)^T(x-\mu_i) \leq (x-\mu_j)^T(x-\mu_j)}$$

K-Nearest Neighbour (KNN)

Let $S = \{ (x_i, \theta_i), i = 1 \text{ to } n, x \in \mathbb{R}^D$
$\theta_i \in \{1 \text{ to } M\}\}$.

Assign x into $\{1 \text{ to } M\}$.

→ Let K be a positive integer. $(k \geq 1)$.

⇒ Calculate k nearest neighbours of x from S.

→ Let k_i be the number of nearest neighbou- from class i.

So, $\sum_{i=1}^{m} k_i = k$.

→ x be in class i,
if $k_i \geq k_j$ for all i, j
$i \neq j$.

⟹ M-2

x be in class 1.

$$p_1 p_1(x) \geq p_2 p_2(x).$$

Assumptions

① $p_i(x)$ follows guassian distribution.

② $\Sigma_1 = \Sigma_2 = \dots \Sigma$

$$\rightarrow p_1 \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-u_1)^T\Sigma^{-1}(x-u_1)\right\}$$

$$\geq p_2 \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} \exp\left\{-\frac{1}{2}(x-u_2)^T\Sigma^{-1}(x-u_2\right.$$

→ Taking log,

$$\rightarrow \log p_1 - \frac{1}{2}(x-u_1)^T\Sigma^{-1}(x-u_1) \geq$$

$$\log p_2 - \frac{1}{2}(x-u_2)^T\Sigma^{-1}(x-u_2)$$

$$\rightarrow \frac{1}{2}\Big[\cancel{x^T\Sigma^{-1}x} - \overbrace{x^T\Sigma^{-1}u_2 - u_2^T\Sigma^{-1}x}^{\text{same}} +$$

$$u_2^T\Sigma^{-1}u_2 - \cancel{x^T\Sigma^{-1}x} + x^T\Sigma^{-1}u_1 +$$

$$u_1^T\Sigma^{-1}x - u_1^T\Sigma^{-1}u_1\Big] \geq \log\frac{p_2}{p_1}$$

$$\rightarrow \frac{1}{2}\left[2x^T\Sigma^{-1}(\mu_1-\mu_2) + \underbrace{\left[\mu_2^T\Sigma^{-1}\mu_2 - \mu_1^T\Sigma^{-1}\mu_1\right]}_{\hookrightarrow\, c}\right] \underbrace{- \log\frac{P_2}{P_1}} \geq 0$$

$$\rightarrow x^T \underbrace{\Sigma^{-1}(\mu_1-\mu_2)}_{W} + c \geq 0$$

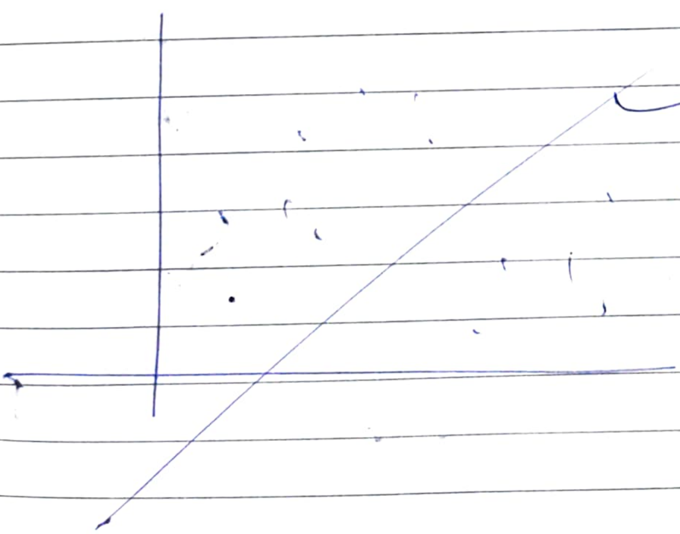
$$\rightarrow [x_1 \;\; x_2 \; \cdots \; x_D] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix} + c \geq 0$$

$$\rightarrow \boxed{w_1x_1 + w_2x_2 + w_3x_3 + \cdots + w_Dx_D + c \geq 0}$$

→ <u>2D</u>

$$w_1x_1 + w_2x_2 + c = 0$$

↳ eqn of line.



→ ~~linear distinct~~ linear discriminant classifier

$\rightarrow$ w is $\perp$ to line.

$\rightarrow x^T w + c = 0$

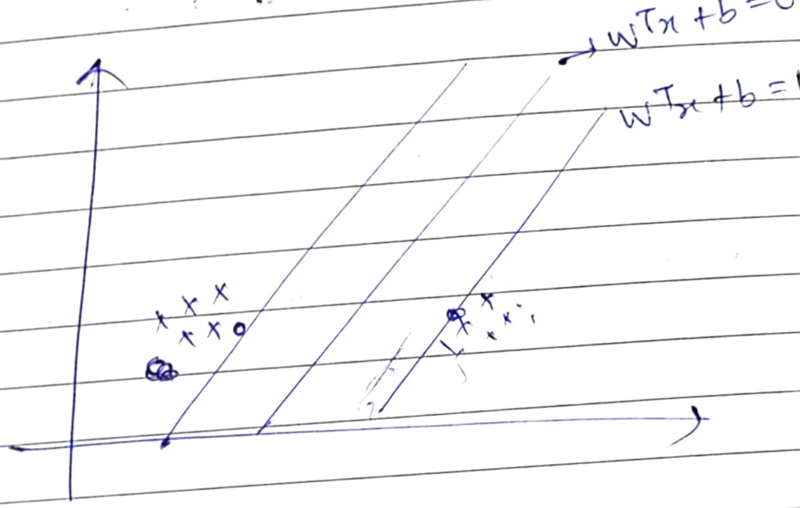
(1) $w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$

14/10/2022

Support Vector Machines (SVM).

$w^T x + b \geq 0 \longrightarrow$ class $C_1 \Rightarrow +1$

$< 0 \longrightarrow$ class $C_2 \Rightarrow -1$



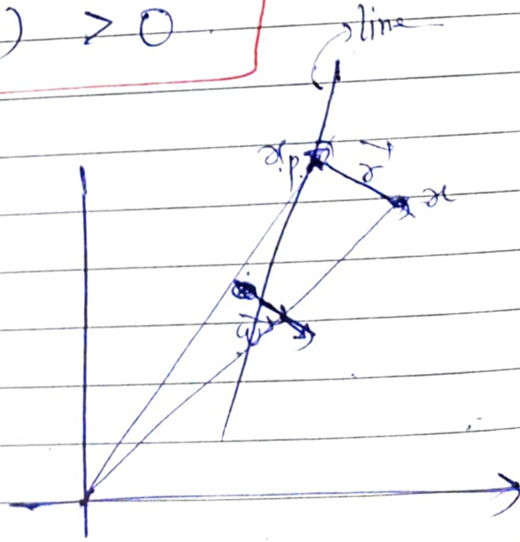
(x_i, y_i) $\qquad y_i \in \{+1, -1\}$

$f(x) = \boxed{y_i (w^T x_i + b) > 0}$

Margin $\rightarrow \delta$

$\vec{\gamma} = \vec{x} - \vec{x}_p$

$\vec{\gamma} \parallel \vec{w}$



Summary

$$\vec{r} = \delta \hat{w} = \delta \frac{\vec{w}}{\|\vec{w}\|}$$

$$\vec{x_p} = \vec{x} - \left(\delta \frac{\vec{w}}{\|\vec{w}\|}\right)$$

$$w^T x_p + b = 0$$

$$\rightarrow w^T x - \delta \frac{w^T w}{\|w\|} + b = 0$$

$$\rightarrow \frac{w^T x + b}{\|w\|} = \delta \qquad \left[\because w^T w = \|w\|^2\right]$$

$$w^T x + b = \delta \|w\|.$$

$$w^T x + b = 1$$

Divide by $\delta \|w\|$.

$$\rightarrow \delta = \frac{w^T x + b}{\|w\|} = \frac{1}{\|w\|}$$

$\rightarrow$ Maximization $\rightarrow$ $\dfrac{2}{\|w\|}$

$\rightarrow$ Minimization $\rightarrow$

$$f(x) = \frac{1}{2}\|w\|$$

$\rightarrow$ Minimization.

$$f(x) = \frac{1}{2}\|w\|^2 = \frac{1}{2} w^T w = \frac{1}{2} w.w$$

Camlin

Minimization of

$$f(w,b) = \frac{1}{2} w.w$$

Constraint

$$g() = y_i(w^T x_i + b) - 1 \geq 0$$

$$\boxed{L(w,b) = \frac{1}{2} w.w - \sum_{i=1}^{n} [y_i(w^T x_i + b) - 1]\alpha_i}$$

Langrange Multiplier

$$\frac{\partial L}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0$$

$$\boxed{\sum_{i=1}^{n} \alpha_i y_i = 0}$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i + \cancel{\sum_{i=1}^{n} \alpha_i} = 0$$

$$\boxed{w = \sum_{i=1}^{n} \alpha_i y_i x_i}$$

$$L(w,b) = \frac{1}{2} \sum_i \sum_j \alpha_i y_i \alpha_j y_j (x_i x_j) -$$

$$\sum_i \sum_j \alpha_i y_i \alpha_j y_j (x_i x_j) -$$

$$\cancel{\sum_{i=1}^{n} (\alpha_i y_i)}^{\,0} b + \sum_{i=1}^{n} \alpha_i = 0$$

$$\boxed{L(w,b) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i}\sum_{j} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)}$$

$\underline{\alpha_i \geq 0}$

1. $\alpha_i = 0 \longrightarrow$ Not Support Vector.
2. $\alpha_i > 0 \longrightarrow$ large value $\Rightarrow$ Support Vector.
3. $\alpha_i >> 0 \longrightarrow$ very high $\longrightarrow$ Noise

$$b = \frac{1}{2}\left[\max\left(\sum_{j,\; y=-1} \alpha_i y_i (x_i \cdot x_j)\right) + \min\left(\sum_{j} \alpha_i y_i (x_i \cdot x_j)\right)_{\text{if } y=1}\right]$$

Test pattern x_p.

$$f(x_p) = \text{Sign}(w^T x_p + b) \begin{cases} \longrightarrow +ve \\ \longrightarrow -ve \end{cases}$$

SVM Linear Classify.

No. of classes > 2.
Multiple SVM's

SVM1 $\longrightarrow$ classifier b/w class 1 and rest of the classes

SVM2 $\longrightarrow$ 1 2

UNIT-2

→ Knowledge Representation ←

⇒ Knowledge.

- It can be defined as body of facts & principles accumulated by human kind or it can be understood as the fact, act or state of knowing.

- knowledge is having familiarity with language, concepts, procedures, rules, ideas, abstractions, places, customs, facts, associations coupled with an ability to use this notions effectively in modelling different aspects of the world.

- The meaning of knowledge is closely related to meaning of intelligence.

- Intelligency requires a ~~p~~ position of access to knowledge.

- A common way to represent external to a computer or ~~man~~ new man is in the form of written language.

- For. eg. 1) Ramu is tall, this expresses a simple fact, which is the attribute possess by a person.

  2) Ramu loves his mother, ~~this~~ it is complex binary relatn b/w 2 persons.

→ There are two entities related to representation of knowledge in AI.

1) facts — the truth in some relevant world, this are things to be represented.

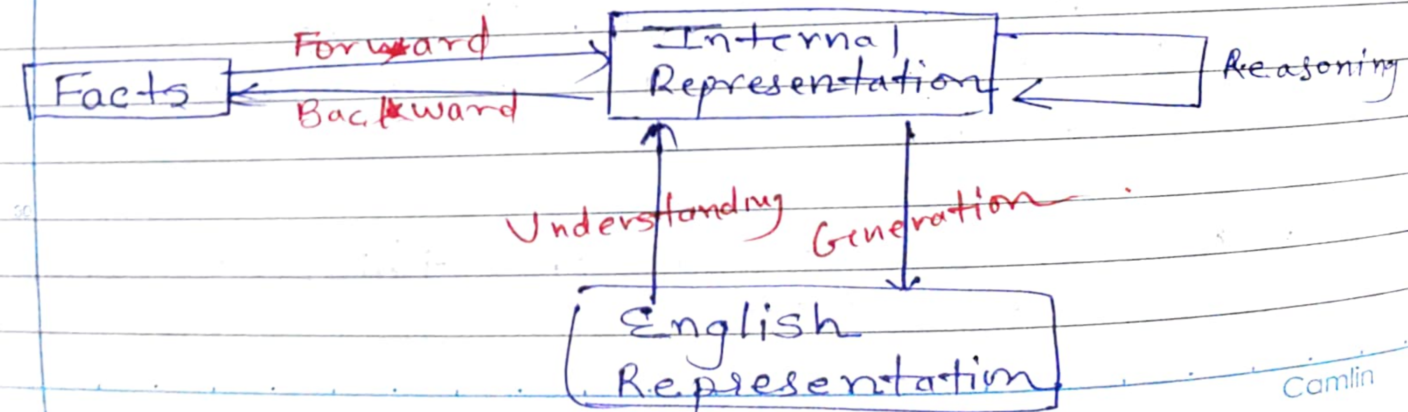
2) Representation of fact — it must be done in some choosen formalism. this are the things we will actually ~~m~~ be able to manipulate → structure

→ The structuring of these entities are done at 2-levels :-

1) Knowledge level — facts including each agents behaviour's and current goals are described at this level.

2) Symbol level — the representation of objects at knowledge level are defined as symbols that can be manupolated by programs.

→ Mapping b/w Facts and Representation



Camlin

→ Consider the statement,
Marcus is a man, using the mathematical logic representation.

1) ⇒ man(Marcus).

→ Suppose, we have a logical representation which says,
all men have legs
⇓
2) ⇒ man(x) ⟶ haslegs(x).

3) haslegs(Marcus). { Deductive mathematical reasoning. }

→ Using backward mapping, we can say that Marcus has legs.

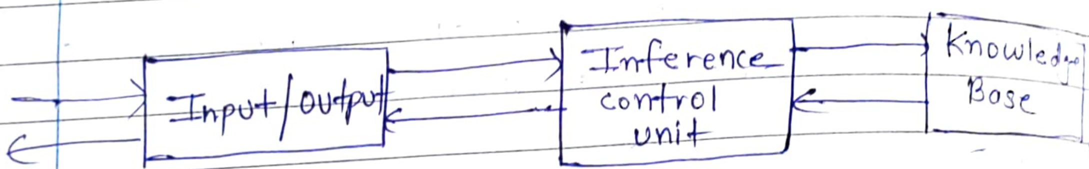
for.eg.
1) All men have legs.
2) Every man has legs.
] → Conclusion
↓
Every man has at least one leg.

Note:-
We must first decide, what facts the sentences represent and then convert those facts into new representation.

12/10/2022
Wednesday

Knowledge based systems

These are the systems that depends on rich base of knowledge to perform difficult tasks. It includes work in vision learning, general problem solving and natural language understanding. The systems get their power from the expert knowledge that has been coded into facts, heuristics and procedure. The knowledge is stored in knowledge-base that is seperate from the control & inferencing componenents. It is possible to add new knowledge or refining existing knowledge without recompiling the controlled programs.



Representation of knowledge

The objective of knowledge representation is to express ~~in a~~ knowledge in a computer tractable form so that it can be used to enable our AI agents to perfom as per the expectation.

The K.R language is defined by two aspects

1) Syntax ⟶
The syntax of a language defines which configurations of the components of the language constitutes valid sentences.

Camlin

i) Semantics :- It defines which facts in the world the sentences refer to & hence the statement the world that each sentence makes.

Approaches to K.R.

- A good system for representation of knowledge in a particular domain should posses following properties :-

i) Representation Adequacy.
→ The ability to represent all the different kind of knowledge that might be needed in that domain.

ii) Inferential Adequacy
→ The ability to manupulate the representat structures in such a way as to derived new structures corresponding to new knowledge infered from old ones.

iii) Inferential Efficiency.
→ The ability to incorporate additional infoⁿ into the knowledge structure, which can be used to focus the attention of the inference mechanism in the most promising directions.

iv) Acquisitional efficiency.
→ The ability to acquire new infoⁿ easily. Ideally the agents should be able to control its own knowledge acquisition.

But, direct insertion of information by knowledge engineer would be acceptable.

Techniques of K.R

1) Simple relational knowledge
It is used in database systems to represent declarative facts.

→ Declarative knowledge is passive knowledge expressed as statements of facts about the world.

2) Inheritable Knowledge
→ It uses the concepts of property inheritance.

3) Inferential knowledge
→ It uses the concept of standard, logical rules of inference.

4) Procedural knowledge
→ It is compiled knowledge related to performance of some tasks.

Fuzzy Logic

1) Fuzzy System
→ Fuzzy Systems include fuzzy logic and fuzzy set theory.

→ Knowledge exists in 2 distinct forms.

(i) Objective knowledge, that exists in mathematical form and is used to represent engineering problems.

(2) Subjective knowledg, that exists in linguistic form and is impossible to quantify.

→ The concept of fuzzy logic is used to coordinate these 2 forms of knowledge in logical way.
Many real world problems have been modeled simulated & replicated with the help of fuzzy systems.

→ Some applications of fuzzy systems are:-
1) Information retrieval
2) Navigation Systems
3) Robot Vision
etc.

→ Fuzzy logic is derived from fuzzy set theory dealing with ~~set theory~~ reasoning that is approximate rather than precisely deduced from classical to valued logic

Declarative Statement that is either true or false is a proposition.
exceptions → commands, questions, exclaimation

Camlin

Truth table banao

Rules of Inference

① Modus Ponens.

$$(p \wedge (p \rightarrow q)) \longrightarrow q$$

② Modus Tollens.

$$(\sim q \wedge (p \rightarrow q)) \longrightarrow \sim p$$

→ In fuzzy logic, the truth values are multi-valued. Such as, absolute true, partially true, absolute false, etc.

⇒ Fuzzy Propositions

— It is a statement, $\tilde{P}$, which acquires a fuzzy truth value $T(\tilde{P})$.

$\tilde{P}$: Ram is honest.

$T(\tilde{P})$: 0.8 → par.
$T(\tilde{P})$: 1 → abs

Fuzzy Connectives

1) Negation → $1 - T(\tilde{P})$

2) Disjunction → $\max\{T(\tilde{P}), T(\tilde{Q})\}$

3) Conjuction → $\min\{T(\tilde{P}), T(\tilde{Q})\}$

4) Conditional (Implication) → $\max\{1 - T(\tilde{P}), T(\tilde{Q})\}$

0.6

$\tilde{P}$ = Many is efficient
$\tilde{Q}$ = Ram is efficient

$T(\tilde{P}) = 0.8$ $\quad T(\tilde{Q}) = 0.65$

Negation of $\tilde{P} \longrightarrow 1 - T(\tilde{P})$

$T(\sim\tilde{P}) \longrightarrow \underline{\underline{0.2}}$

Fuzzy Quantifiers

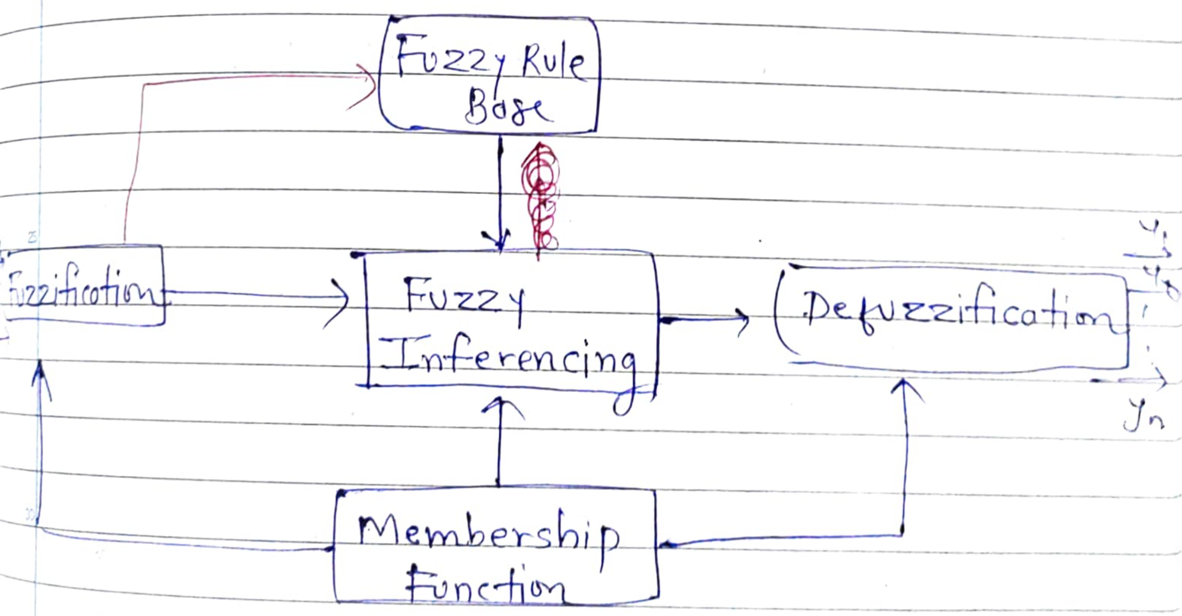
In crisp logic
↳ ordinary set

In

There are 2 classes of fuzzy quantifiers:
1) absolute quantifiers → round about 200
2) relative quantifiers → almost 200

~~much greater than 200~~

Fuzzy Systems



⇒ Fuzzy System Agents.

1) Input Vector → This are crisp values. This are transformed into fuzzy sets in fuzzification block

$$X = [x_1, x_2 \ldots x_n]^T$$

2) Output Vector → It comes out of the de-fuzzification block which ~~out~~ transforms an output fuzzy set back to crisp values.

3) Fuzzification — It is a process of transforming crisp values into grades of membership for linguistic term such as 'far', 'small', 'near' of fuzzy sets.

4) Fuzzy Rule base → It is a collection of propositions containing linguistic variables.

IF (x is A) (Y is B) THEN (z is C)

where A, B, C are linguistic terms.
x, y, z → variables.

5) Membership function.
It provides a measure of the degree of similarity of elements in the Universe of discourse, U, to fuzzy set.

6) Fuzzy Inferencing
It combines the fact obtained from fuzzification with the rule base and conducts the ~~rule~~ reasoning.

Camlin

7) It translates the results back the the

→ The typical defuzzication methods are

1) centroid methods
2) centre of sums.
3) Mean of maxima .

Self Paced Assng.

Write Short Note on knowledge base agent & read about example on wumpus world

17/10/21 →

First Order Logic.

→ First order logic is another way of representing knowledge.

→ FOPL.

→ It is an extension to propositional ~~kno~~ logic.

→ It is a powerful language that develops more information about the objects in a easy way, but can also express the relⁿ b/w those objects.

Syntax of FOL.

→ Some basic elements of FOL are :-

1) constants
2) variables
3) predicates.
4) functions
5) connectives
6) equality.
7) quantifiers.

Atomic Sentences

→ Most basic sentences of FOL, and are formed from a predicate symbol followed by parenthesis with a sequence of terms.

→ represented as :-

predicate (term1 term2, ... term n).

→ for eg
Hari and Raghu are brothers.

⟹ brothers (Hari, Raghu).

→ Tommy is a dog.

dog (Tommy).

Complex Sentences

It is made by combining atomic sentences using connectives.

FOL statements are divided into 2 parts.
1) Subject ⟶ main part
2) Predicate ⟶ It is a relⁿ which binds atoms together in a statement

⟶

Quantifiers

It is a language which generates quantification

They are the symbols that permit to determine the identity of variable in terms of range and scope in logic expressions.

Universal Quantifiers ($\forall$)

$\longrightarrow$ for all, for each

($\Rightarrow$)

Existential Quantifiers ($\exists$)

$\longrightarrow$ for some, there exist.

($\wedge$)

It is a symbol of logical representation, which specifies that the statements within its range is true.

For everything and every instance it is true

It is represented as $\forall$, $\Rightarrow$

It is a type of quantifier which expresses the statement in its scope is true for atleast one instance of something.

$\longrightarrow$ All men ~~b~~ drink coffee.

$x \longrightarrow$ variable

x_1 ~~drink~~ coffee $\wedge$ x_2 coffee, $\wedge$ so on.

$\longrightarrow \forall x \ man(x) \longrightarrow drink(x, coffee)$

Camlin

Eg. All birds fly.
fly(bird)
→ $\forall x$ bird$(x) \longrightarrow$ fly(x).

Eg. Every man respects his parents.
respects(x, y)
$\forall x$ man$(x) \longrightarrow$ respects$(x$, parents$)$

Eg. Some boys are intelligent.

x_1 is intelligent $\vee$ x_2 is intelligent & so on

→ $\exists x$: boy$(x) \wedge$ intelligent(x)

Inferences in FOL

It is help to deduce new facts & sentences from existing ones.

Terminologies.

1) Substitution :- It is fundamental operⁿ perform on terms or formula.
It occurs in all systems in FOL.

$F[a/x]$ → a is substituted for x.
constant, variable

2) Equality :-

brother(john) = smith

Inference rule for quantifiers

1) Universal generalization :-
It is a valid inference rule which states that if premise p(c) is true for any arbitrary element c in the universe of discourse, then for all x, p(x) is conclusion.

2) Universal Instantiation.
It is also called universal elimination. It can be applied multiple times to add new sentences.
We can infer any sentence p(c), by substituting a ground term c from all x, p(x) for any object in the universe of discourse.

for eg every person likes icecream
↓
abhay likes icecream

3) Existential elimination.
It is a valid inference rule in FOL, and can be applied once, to replace the existential sentences.
One can infer p(c), from the formula in the form

$\exists x \; P(x)$

$\therefore \; p(c)$.

1) Existential
If their is some element c, in the universe of discourse which has a property p, then we can infer that there exist something in universe there exist property p.

(e.g.) A man got good marks in AI.
↓
some one got good marks in AI.

Unification

It is all about looking the expressions look identicle.

$P(x, F(g))$
$P(y, F(z))$ } Identicle if $x = y$ and $g = z$.

→ Cond^ns for unification.

① Predicate symbol must be same atoms or expressions with diff predicate symbol can never be unified

② No. of arguments in both expressions must be identicle.

③ Unification will fail, if there are 2 similar variables present in some expression

Camlin

Unification Algorithm

Unify (L_1, L_2)

Step 1) If L_1 or L_2 is a variable or constant then

a) if L_1 and L_2 are identicle return nil

b) else if L_1 is a variable then

i) if L_1 occurs in L_2 then return fail

ii) else return (L_2/L_1)

c) else if L_2 is a variable then

i) if L_2 occurs in L_1 then return fail.

ii) else return (L_1/L_2).

d) else return fail.

Step 2 If initial predicate symbol in L_1 and L_2 are not identicle, then return fail

Step 3 If L_1 and L_2 have diff. no. of arguments then return fail.

Step 4) Set subst to ~~nil~~ NIL.

Step 5) Loop

for i=1 to number of arguments in L_1

a) call unify with ith arg. of L_1 and ith arg. of L_2 in S.

b) if S = fail then return fail.

~~c) If S != NULL then re~~

then

(i) apply S_2 to the remainder of both L_1 and L_2

(2) SUBST = APPEND (S, SUBST).

Step 6) return SUBST.

Implementation of Unification Algorithm

- Initialise the substitution set to be empty.
- Recursively unify ~~all~~ atomic sentences.
  (a) check ~~for~~ identicale expsn match.
  (b) if one exprn is a variable V_i and other is a term T_i, which doesnot contain V_i.
  then
  (i) substitute $[T_i/V_i]$ in existing substitutions
  (ii) add $[T_i/V_i]$ to the substitution set list.
  (iii) If both the expressions are functions then function names must be similar and no. of arg. must be same in both expsn.

eg:- $P(x, g(x))$

(1) $P(z, y)$
→ possible.

(2) $P(z, g(z))$
yes.

(3) P(prince, f(psme))

No