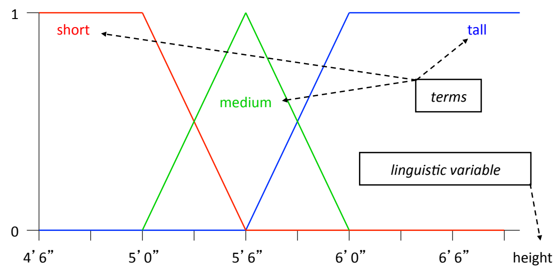
G53FUZ
Fuzzy Sets and Systems

Linguistic Variables

Jon Garibaldi
Intelligent Modelling and Analysis Research Group

Informal Definition

- A *linguistic variable* is a collection of fuzzy sets representing linguistic terms of a concept



Formal Definition

- A linguistic variables is characterised by a quintuple (*X*, *T*, *U*, *G*, *M*)
  - *X* the name of the variable (e.g. *height*)
  - *T* the set of terms, each being a fuzzy variable (e.g. *short*, *medium*, *tall*)
  - *U* the universe of discourse common to all terms, which is associated with a *base variable u*
  - *G* a syntactic rule (grammar) for generating composite terms (e.g. *very short or very tall*)
  - *M* a semantic rule for associating each term with its meaning (fuzzy set)

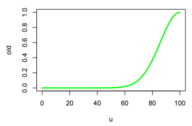
Example

- Consider a linguistic variable named *Age*
  - *X* = *Age*
- Defined over a universe of discourse
  - *U* = [0, 100]
- The term-set *T* associated with *Age* may be
  - *T* = *young* + *very young* + *not young* + *middle-aged* + *not middle-aged* + *old* + *very old* + *not old* + *young* or *middle-aged* + *not very old* + …
  - some terms are *atomic* (*young*)
  - some terms are *composite* (*not young*)

Meaning of Terms

- The terms are names (words)
  - *young*, *middle-aged*, *old*, …
- The *meaning of terms* are fuzzy sets
  - $M(young) = \int_0^{100} e^{-\frac{u^2}{20^2}}$

  - $M(old) = \int_0^{100} e^{-\frac{(100-u)^2}{20^2}}$



Grammar

- Zadeh’s original definition has the concept of grammar, *G*, which generates the set of terms *T*
  - *T* → *young*
  - *T* → *middle-aged*
  - *T* → *old*
  - *T* → *not T*; *T* → *T and T*; *T* → *T or T*
  - *T* → *very T*; *T* → *somewhat T*
  - *T* → (*T*)
  - note that this simple grammar allows terms such as *not very very not young*

A Better Grammar?

- Zadeh gives the production system
  - *T* → *A*
  - *T* → *T or A*
  - *A* → *B*
  - *A* → *A and B*
  - *B* → *C*
  - *B* → *not C*
  - *C* → (*T*)
  - *C* → *D*
  - *C* → *E*
  - *D* → *very D*
  - *E* → *very E*
  - *D* → *young*
  - *E* → *old*
  - this tries to restrict the terms to give a better match to natural language: try it!

Informal Grammars

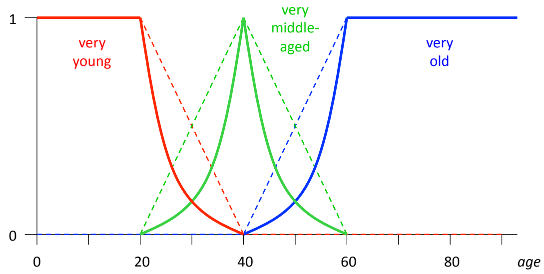
- In practice, the concept of formal grammar is very rarely used
- There is usually an informal idea that there is a set of terms
  - *young, middle-aged, old*
- And then standard operators can be applied
  - *not, and, or*
- Sometimes (very rarely) hedges are also used
  - *very, somewhat*

Hedges

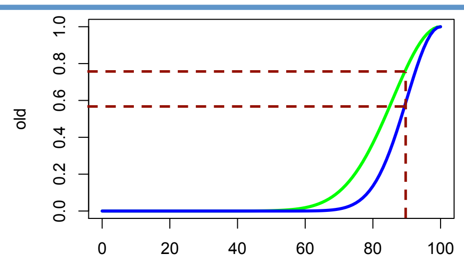
- A hedge is a qualifying word added to a term to indicate a minor modification of the usual meaning of the term
- In English, common hedges are
  - very, extremely
  - rather, quite, slightly
  - somewhat, more or less

Concentration

- Squaring a membership function makes it more concentrated ≡ '*very*'



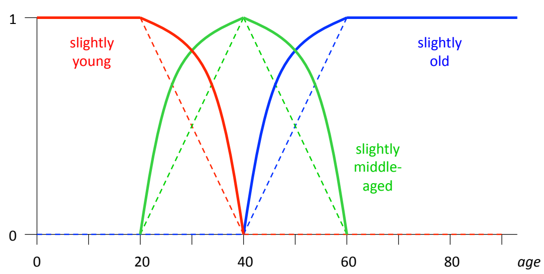
Example 'Very Old'



- old[90]= 0.74
- very_old[90]= 0.54

Dilation

- Square-rooting a membership function makes it less concentrated (more dilated) ≡ '*slightly*'

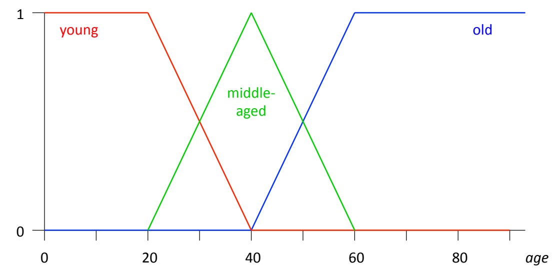


Membership Functions

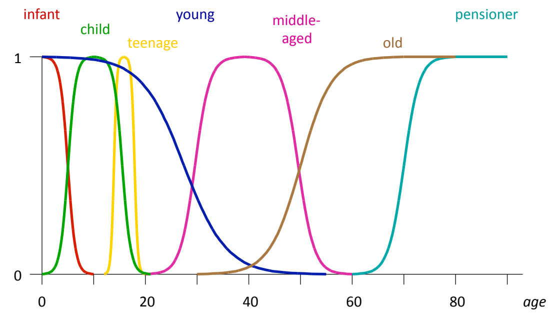
- Often, the meaning of terms such as *M*(*young*) are defined by functions
  - not necessarily the case
  - could be defined by enumeration (look-up table)
- When there is a function it is called a ***membership function***
  - in this case, *M* can be thought of as the set of membership functions relating the *names of the terms* to the *meaning of the terms*
- Usually written as *young* = …

Terms

- The number and shape of terms may be application dependent

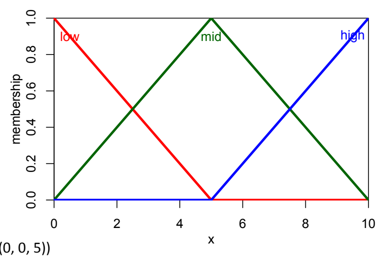


Alternative Terms



Common Membership Functions

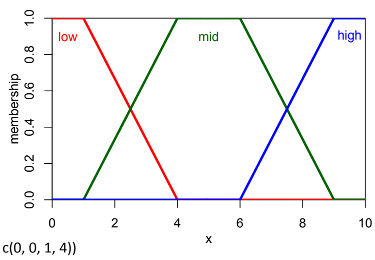
- Triangular
  - usually by three params (*left*, *centre*, *right*)



low= trimf(x, c(0, 0, 5))

Common Membership Functions

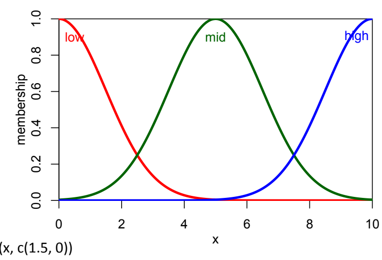
- Piece-wise linear / trapezoidal
  - shoulders by three params (*left*, *centre*, *right*)
  - trapezoids by four (*lb*, *lt*, *rt*, *rb*)



low= trapmf(x, c(0, 0, 1, 4))

Common Membership Functions

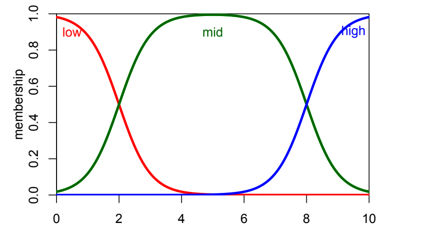
- Gaussian
  - two params (*centre*, *standard deviation*)
  - note in R/MATLAB it's (*standard deviation*, *centre*)



low= gaussmf(x, c(1.5, 0))

Common Membership Functions

- Sigmoids
  - two params (*slope*, *half-point*)
  - double formed by diff/product of two sigmoids



low= sigmf(x, c(-2, 2))   mid= dsigmf(x, c(2,2,2,8)) = psigmf(x, c(2,2,-2,8))

Deriving Terms

- Questions arise when designing an application
  - how many terms should there be?
  - what shape should they be?
  - how much overlap should there be?
- Methods for deriving terms
  - do a survey
  - ask domain experts
  - build a system which 'learns' shapes from data
  - guess!?

Guidelines

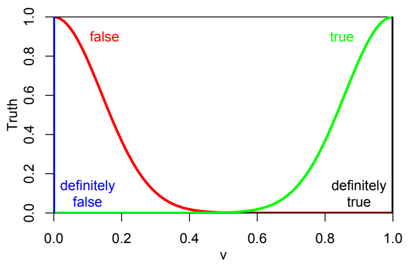
- There are a number of heuristics (rules-of-thumb) that can be applied to membership functions of terms in a linguistic variable
  - the terms should span the universe of discourse
  - the terms should not overlap too much
    - terms should overlap at around 0.5 membership
  - the number of terms should be small (≤ 7)
  - all terms are normal
  - all terms are convex
  - there should be an odd number of terms

Linguistic Truth

- Now we have the concept of a linguistic variable, it is possible to have linguistic truth
  - *X* = *Truth*
  - *U* = *V*= [0, 1]
  - *T* = true + not true + very true + somewhat true + definitely true + … + false + not false + very false + somewhat false + definitely false + …
- So, we can now represent and systematise statements such as *"that's not very true!"*

Truth Value Meanings

- *false* = $\int_0^1 e^{-\frac{v^2}{0.2^2}}$
- *true* = $\int_0^1 e^{-\frac{(1-v)^2}{0.2^2}}$
- *definitely false* = 1/0
- *definitely true* = 1/1



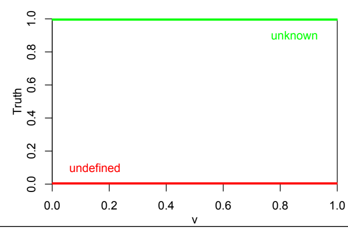
Special Terms (Level Sets)

- The term *undefined* can be defined (!) as

$$\theta = \int_0^1 0 / v$$

- The term *unknown* can be defined as

$$? = \int_0^1 1 / v$$



Fuzzy Logic?

- With these concepts, it is possible to apply fuzzy operations to derive results, e.g.
  - *true* = 0.5/0.7 + 0.7/0.8 + 0.9/0.9 + 1.0/1.0
  - *false* = 1.0/0.0 + 0.9/0.1 + 0.7/0.2 + 0.5/0.3
  - *very true* = (*true*)^2
    = 0.01/0.6 + 0.09/0.7 + 0.25/0.9 + 0.49/0.9 + 1.0/1
- However, the standard intersection and union operators produce unintended results
  - *false* ∩ *true* = 0 / *v* = *undefined*

The Extension Principle

- Zadeh asserted a basic identity which allows a relationship from one domain to another to be extended into fuzzy domains
  - f is a mapping (function) from U to V
  - $A = \mu_1 u_1 + \ldots + \mu_n u_n$
- Then, the *extension principle* asserts that
  - $f(A) = f(\mu_1 u_1 + \ldots + \mu_n u_n) \equiv \mu_1 f(u_1) + \ldots + \mu_n f(u_n)$
- Note the difference between e.g. *small*^{2} and the hedge *very small*

Fuzzy Logic

- By applying the extension principle, it is possible to create new connectives which correspond better to logical meanings
- This then provides a full framework for representing and manipulating linguistic truth values which arise in everyday usage
  - includes various terms such as 'unknown', 'undefined' and 'undecidable'
  - can represent differences between
    - definitely true, very true, true-ish, etc.

Logical Connectives

$$v(A) = \alpha_1 / v_1 + \ldots + \alpha_n / v_n$$

$$v(B) = \beta_1 / w_1 + \ldots + \beta_m / w_m$$

- Logical and

$$\begin{aligned} v(A \text{ and } B) &= v(A) \wedge v(B) \\ &= (\alpha_1 / v_1 + \ldots + \alpha_n / v_n) \wedge (\beta_1 / w_1 + \ldots + \beta_m / w_m) \\ &= \sum_{i,j} (\alpha_i \wedge \beta_j) / (v_i \wedge w_j) \end{aligned}$$

- Logical or

$$\begin{aligned} v(A \text{ or } B) &= v(A) \vee v(B) \\ &= (\alpha_1 / v_1 + \ldots + \alpha_n / v_n) \vee (\beta_1 / w_1 + \ldots + \beta_m / w_m) \\ &= \sum_{i,j} (\alpha_i \wedge \beta_j) / (v_i \vee w_j) \end{aligned}$$

Logical Examples

- Recall
  - *true* = 0.5/0.7 + 0.7/0.8 + 0.9/0.9 + 1.0/1.0
  - *false* = 1.0/0.0 + 0.9/0.1 + 0.7/0.2 + 0.5/0.3
- *v*(*true* and *false*)

∑ min(0.5,1.0)/min(0.7,0.0) + min(0.5,0.9)/min(0.7,0.1) +
min(0.5,0.7)/min(0.7,0.2) + min(0.5,0.5)/min(0.7,0.3) +
min(0.7,1.0)/min(0.7,0.0) + min(0.7,0.9)/min(0.7,0.1) +
min(0.7,0.7)/min(0.7,0.2) + min(0.7,0.5)/min(0.7,0.3) +
min(0.9,1.0)/min(0.7,0.0) + min(0.9,0.9)/min(0.7,0.1) +
min(0.9,0.7)/min(0.7,0.2) + min(0.9,0.5)/min(0.7,0.3) +
min(1.0,1.0)/min(0.7,0.0) + min(1.0,0.9)/min(1.0,0.1) +
min(1.0,0.7)/min(0.7,0.2) + min(1.0,0.5)/min(1.0,0.3)

Logical Examples

∑ 0.5/0.0 + 0.5/0.1 + 0.5/0.2 + 0.5/0.3 +
0.7/0.0 + 0.7/0.1 + 0.7/0.2 + 0.5/0.3 +
0.9/0.0 + 0.9/0.1 + 0.7/0.2 + 0.5/0.3 +
1.0/0.0 + 0.9/0.1 + 0.7/0.2 + 0.5/0.3

= 1.0/0.0 + 0.9/0.1 + 0.7/0.2 + 0.5/0.3

= *false*

- *i.e.. true* and *false* = *false*, as expected ☺
- Other rules of crisp logic are adhered to
  - try evaluating '*true* or *false*'
  - but new relationships are also provided for
    - e.g. '*true* and *unknown*', etc.

Linguistic Probability

- Probability can also be represented with a linguistic variable
  - X = *Probability*
  - U = [0, 1]
  - T = likely + not likely + unlikely + very likely + …
    + probable + improbable + …
    + possible + impossible + …
    + some chance + no chance + …
- Similar construction to fuzzy logic

Summary

- Lecture summary
  - linguistic variables are the formal devices for the definition of variables that take linguistic terms
  - terms are defined by fuzzy membership functions
    - fuzzy sets
  - full fuzzy logic allows the representation and manipulation of linguistic logic terms
    - full fuzzy logic is not commonly seen in practice
- Next lecture
  - fuzzy inference