Chapter 6
Development of Membership Functions

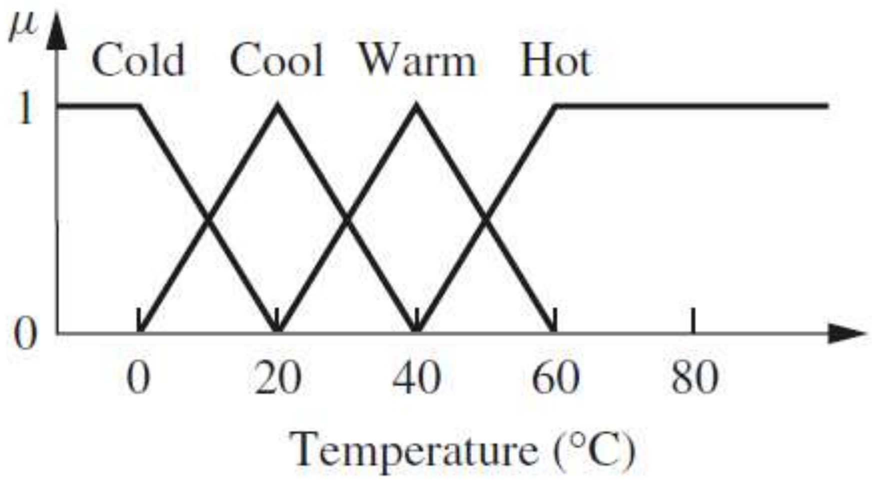
Dr. Wafa' H. AlAlaween
wafa.alalaween@gmail.com

Membership Value Assignments

- The assignment process can be intuitive or based on some algorithmic or logical operations.
- Common methods are:
  1. Intuition
  2. Inference
  3. Rank ordering
  4. Neural networks
  5. Genetic algorithms
  6. Inductive reasoning

Intuition

- The membership values or functions can be derived from the capacity of humans to develop them through their own innate intelligence and understanding.
- Example: Develop fuzzy membership functions for the temperature.
  - Very cold
  - Cold
  - Normal
  - Hot
  - Very hot



Inference

- Knowledge is utilised to perform deductive reasoning.
- Example: Let U be the universe of triangles, where the inner angles are A, B and C.

$\underset{\sim}{\mathrm{I}}$ Approximate isosceles triangle

$\underset{\sim}{\mathrm{R}}$ Approximate right triangle

$\underset{\sim}{\mathrm{IR}}$ Approximate isosceles *and* right triangle

$\underset{\sim}{\mathrm{E}}$ Approximate equilateral triangle

$\underset{\sim}{\mathrm{T}}$ Other triangles.

$$\mu_{\underset{\sim}{\mathrm{I}}}(A, B, C) = 1 - \frac{1}{60^\circ} \min(A - B, B - C)$$

$$\mu_{\underset{\sim}{\mathrm{R}}}(A, B, C) = 1 - \frac{1}{90^\circ} |A - 90^\circ|$$

$$\underset{\sim}{\mathrm{IR}} = \underset{\sim}{\mathrm{I}} \cap \underset{\sim}{\mathrm{R}},$$

$$\mu_{\underset{\sim}{\mathrm{E}}}(A, B, C) = 1 - \frac{1}{180^\circ}(A - C)$$

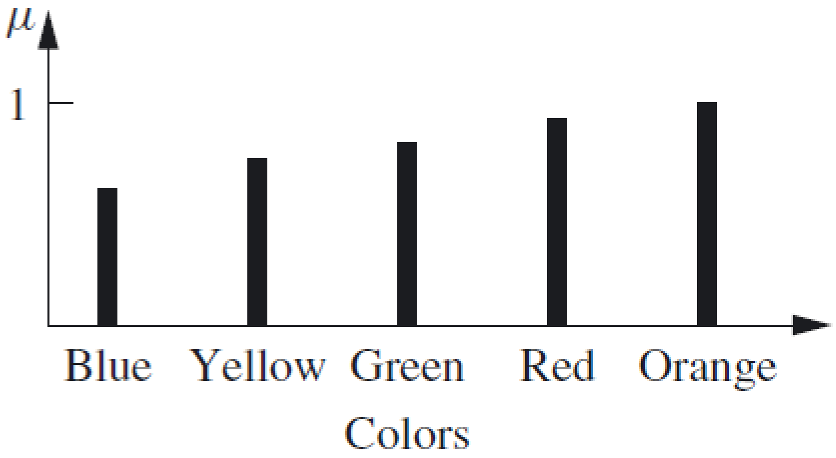
Rank Ordering

- Preference is determined by pairwise comparisons, and these determine the ordering of the membership.
- Example: Suppose 1000 people respond to a0about their pairwise preferences among five colours, X = {red, orange, yellow, green, blue}.

	Number who preferred					Total	Percentage	Rank order
	Red	Orange	Yellow	Green	Blue			
Red	–	517	525	545	661	2 248	22.5	2
Orange	483	–	841	477	576	2 377	23.8	1
Yellow	475	159	–	534	614	1 782	17.8	4
Green	455	523	466	–	643	2 087	20.9	3
Blue	339	424	386	357	–	1 506	15	5
Total						10 000		

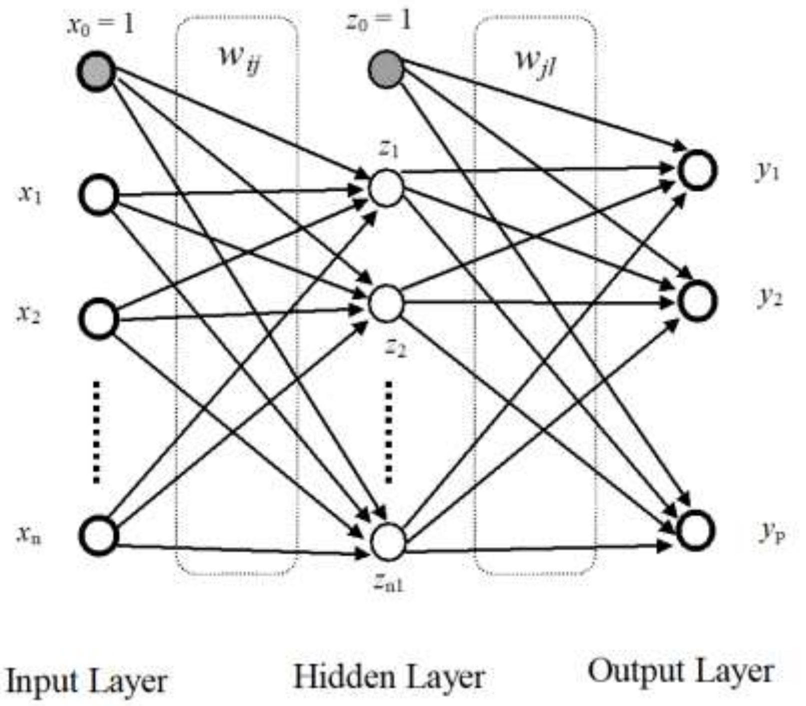
Continue…

- Membership function for the best colour



Neural Networks

- A neural network: is a technique that builds an intelligent system by simulating the biological neural network.



$$z_j(k) = f_j\left(\sum_{i=1}^{n} w_{ij} x_i(k) + b_j\right), \quad j = 1,2,\ldots,n_1; \quad k = 1,2,\ldots$$

$$y_l(k) = f_l\left(\sum_{j=1}^{n_1} w_{jl} z_j(k) + b_l\right); \quad l = 1,2,\ldots,p; \quad k = 1,2,\ldots$$

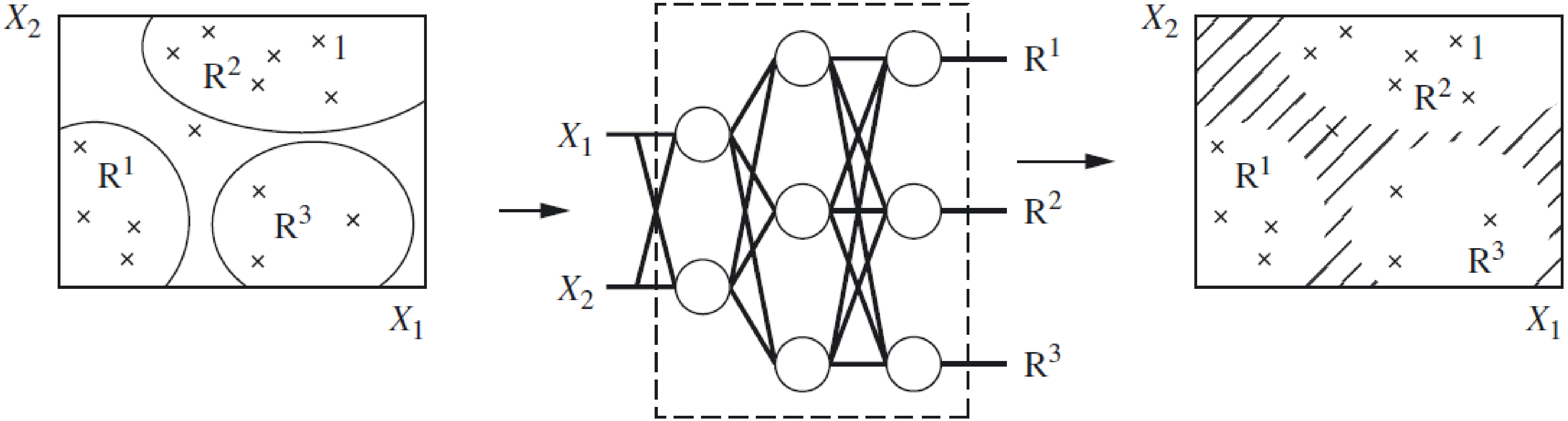
$$E(k) = \frac{1}{2}\sum_{l=1}^{p}\left(y_l(k) - y_l^t(k)\right)^2$$

$$w_{jl}(k+1) = w_{jl}(k) - \alpha \nabla_{w_{jl}} E(k)$$

Distributing the error using back-propagation technique

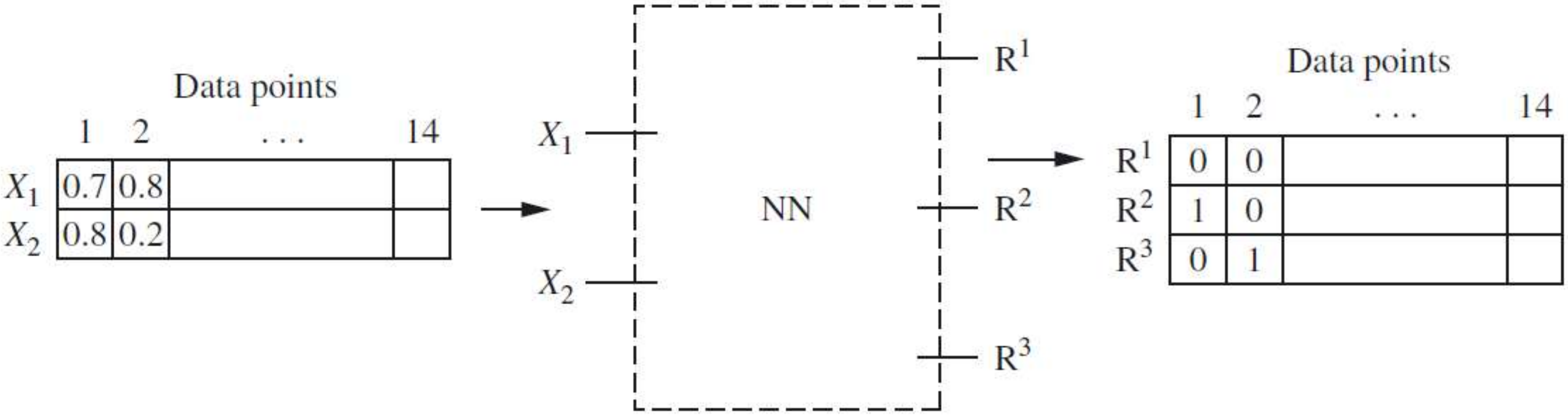
Continue…

- Determining the membership function



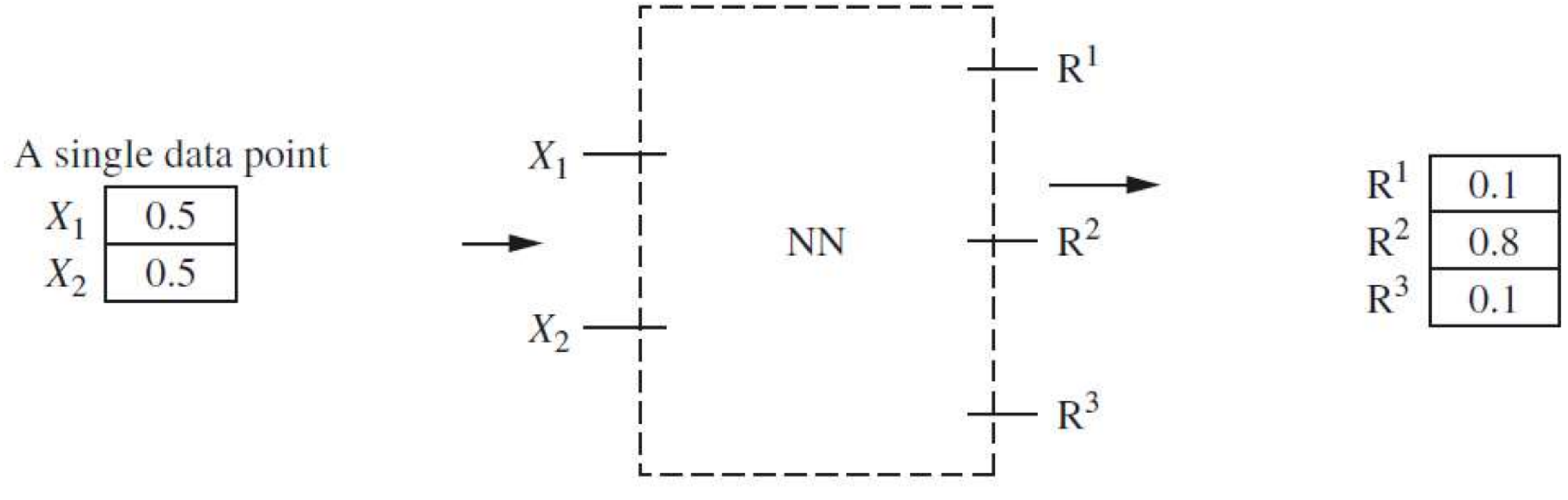
(a) (b) (c)

Continue…



(d) (e) (f)

Continue…



(g) (h) (i)

Genetic Algorithms

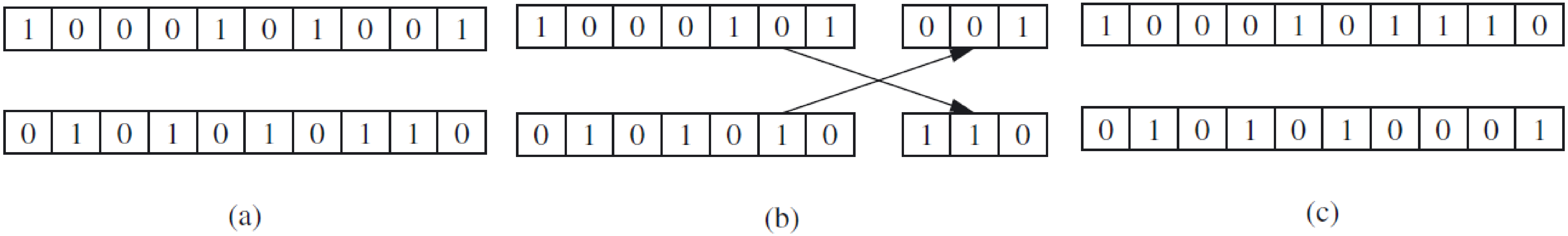
- Genetic algorithms use the concept of Darwin’s theory of evolution; “survival of the fittest”.
- New breeds or classes come into existence through the processes of reproduction, crossover, and mutation among existing organisms.
- The algorithms procedure can be summarized as follows:
  - Different possible solutions are created.
  - They are then tested for their performance.
  - Among all of them, a fraction of the good solutions is selected, and the others are eliminated.
  - The selected solutions undergo the processes of reproduction, crossover, and mutation to create a new generation of possible solutions.

Continue…

- In a genetic algorithm, the parameter set is coded as a finite string, which is presented as a combination of zeros and ones.
- For example, the number 7 requires a 3-bit string, that is, $2^3 - 1 = 7$, and the bit string would look like “111”.
- So the number 10 would look like: “1010”.

Continue…

- Reproduction is the process by which strings with better fitness values receive correspondingly better copies in the new generation, to ensure that better solutions persist and contribute to better offspring (new strings).
- Crossover is the process in which the strings are able to mix and match their desirable qualities in a random fashion.



Continue…

- Mutation is the process by which the value at a certain string location is changed; if there is a one originally at a location in the bit string, it is changed to a zero, or vice versa.
- Mutation takes place very rarely, on the order of once in a thousand bit string locations.

Genetic Algorithms: Example

- Using the data provided in the table below, perform a line fit ($y=C_1x+C_2$).

Data number	x	y′
1	1.0	1.0
2	2.0	2.0
3	4.0	4.0
4	6.0	6.0

Continue…

(1) String number	(2) String	(3) C_1	(4) C_1 (binary)	(5) C_2	(6) C_2 (binary)	(7) y_1	(8) y_2	(9) y_3	(10) y_4		(11) $f(x) = 400 - \Sigma(y_i - y_i')^2$	(12) Expected count = f/f_{av}	(13) Actual count
1	000111 010100	7	−1.22	20	0.22	−1.00	−2.22	−4.66	−7.11		147.49	0.48	0
2	010010 001100	18	0.00	12	−0.67	−0.67	−0.67	−0.67	−0.67		332.22	1.08	1
3	010101 101010	21	0.33	42	2.67	3.00	3.33	5.00	4.67		391.44	1.27	2
4	100100 001001	36	2.00	9	−1.00	1.00	3.00	3.67	11.00		358.00	1.17	1
										Sum	1229.15		
										Average	307.29		
										Maximum	391.44		

$$C_i = C_{\min} + \frac{b}{2^L - 1}(C_{\max_i} - C_{\min_i})$$

The minimum and the maximum values are -2 and 5, respectively. L=6 bits and b is the number in the decimal form.

To convert the problem into a maximization one with a cut-off value equals to 0.8.

Number of copies

Continue…

(1) Selected strings	(2) New strings	(3) C_1	(4) C_1 (binary)	(5) C_2	(6) C_2 (binary)	(7) y_1	(8) y_2	(9) y_3	(10) y_4		(11) $f(x) = 400 - \Sigma(y_i - y_i')^2$	(12) Expected count = f/f_{av}	(13) Actual count
0101|01 101010	010110 001100	22	0.44	12	−0.67	−0.22	0.22	1.11	2.00		375.78	1.15	1
0100|10 001100	010001 101010	17	−0.11	42	2.67	2.56	2.44	2.22	2.00		380.78	1.17	2
010101 101|010	010101 101001	21	0.33	41	2.56	2.89	3.22	3.89	4.56		292.06	0.90	1
100100 001|001	100100 001010	36	2.0	10	−0.89	1.11	3.11	7.11	11.11		255.73	0.78	0
										Sum	1304.35		
										Average	326.09		
										Maximum	380.78		

Genetic Algorithms: MF

- Membership functions and their shapes are assumed for various variables defined for a problem. They are then coded as bit strings.

Example

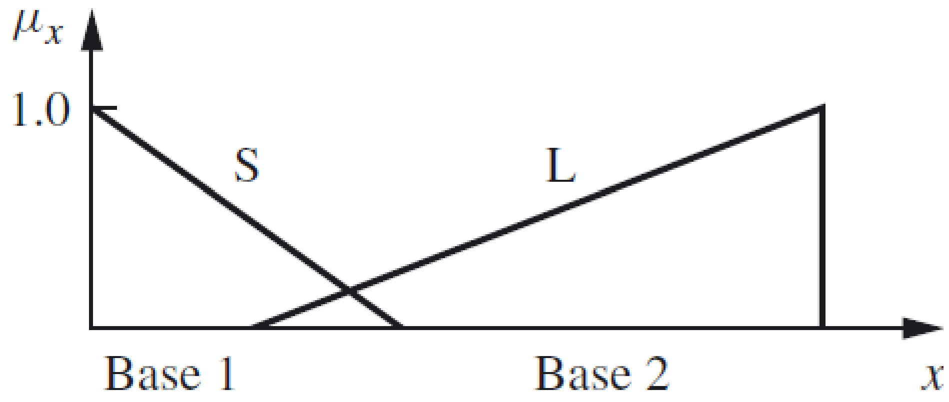
- Let us consider that we have a single-input (x), single-output (y) system with input–output values as shown below:

x	1	2	3	4	5
y	1	4	9	16	25

x	S	L
y	S	VL

Continue…

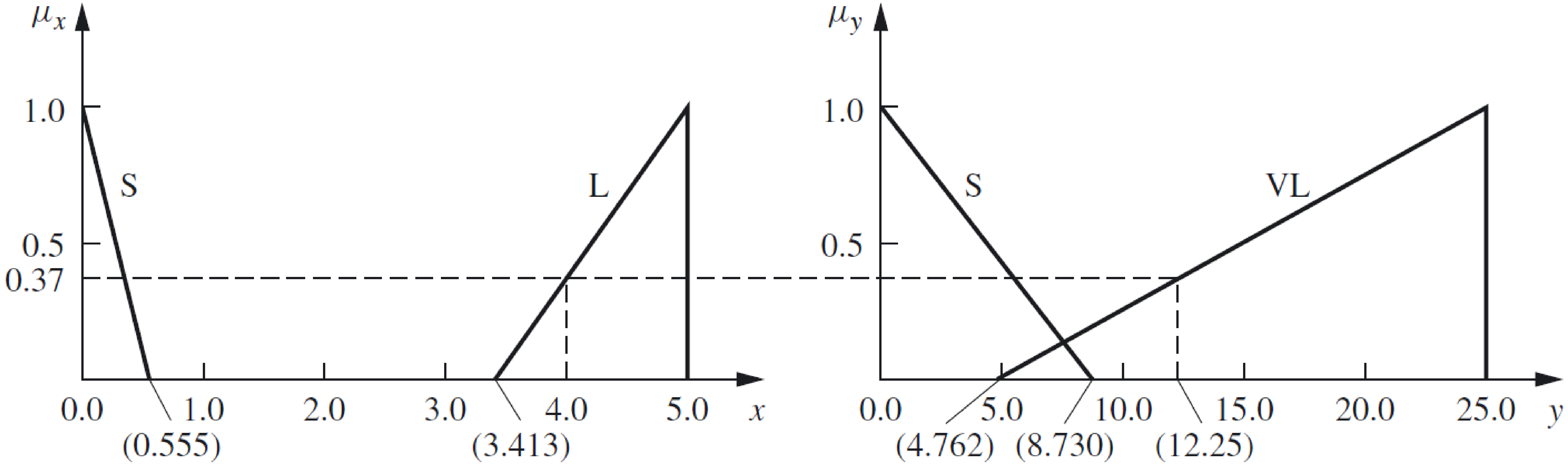
- We assume that the range of the variable x is [0, 5] and that of y is [0, 25].
- Membership function:



Continue…

String number	(1) String	(2) Base 1 (binary)	(3) Base 2 (binary)	(4) Base 3 (binary)	(5) Base 4 (binary)	(6) Base 1	(7) Base 2	(8) Base 3	(9) Base 4	(10) y' $(x = 1)$	(11) y' $(x = 2)$	(12) y' $(x = 3)$	(13) y' $(x = 4)$	(14) y' $(x = 5)$	(15) $1000 - \Sigma(y_i - y_i')^2$	(16) Expected count = f/f_{av}	(17) Actual count
1	000111 010100 010110 110011	7	20	22	51	0.56	1.59	8.73	20.24	0	0	0	12.25	25	887.94	1.24	1
2	010010 001100 101100 100110	18	12	44	38	1.43	0.95	17.46	15.08	12.22	0	0	0	25	521.11	0.73	0
3	010101 101010 001101 101000	21	42	13	40	1.67	3.33	5.16	15.87	3.1	10.72	15.48	20.24	25	890.46	1.25	2
4	100100 001001 101100 100011	36	9	44	35	2.86	0.71	17.46	13.89	6.98	12.22	0	0	25	559.67	0.78	1
														Sum	2859.18		
														Average	714.80		
														Maximum	890.46		

Continue…



μ_x
1.0
0.5
0.37
S
L
0.0
1.0
2.0
3.0
4.0
5.0
x
(0.555)
(3.413)
μ_y
1.0
0.5
S
VL
0.0
5.0
10.0
15.0
20.0
25.0
y
(4.762)
(8.730)
(12.25)

Continue…

(1) Selected strings	(2) New Strings	(3) Base 1 (binary)	(4) Base 2 (binary)	(5) Base 3 (binary)	(6) Base 4 (binary)	(7) Base 1	(8) Base 2	(9) Base 3	(10) Base 4	(11) y' $(x = 1)$	(12) y' $(x = 2)$	(13) y' $(x = 3)$	(14) y' $(x = 4)$	(15) y' $(x = 5)$	(16) $1000 - \Sigma(y_i - y_i')^2$	(17) Expected count = f/f_{av}	(18) Actual count
000111 0101|00 010110 110011	000111 010110 001101 101000	7	22	13	40	0.56	1.75	5.16	15.87	0	0	0	15.93	25	902.00	1.10	1
010101 1010|10 001101 101000	010101 101000 010110 110011	21	40	22	51	1.67	3.17	8.73	20.24	5.24	5.85	12.23	18.62	25	961.30	1.18	2
010101 101010 001101 101000	010101 101010 001101 10|0011	21	42	13	35	1.67	3.33	5.16	13.89	3.1	12.51	16.68	20.84	25	840.78	1.03	1
100100 001001 101100 100011	100100 001001 101100 10|1000	36	9	44	40	2.86	0.71	17.46	15.87	6.11	12.22	0	0	25	569.32	0.70	0
														Sum	3 273.40		
														Average	818.35		
														Maximum	961.30		