MONASH University

Information Technology

FIT5186 Intelligent Systems

Lecture 9

Other Intelligent Techniques

Learning Objectives

- Understand
  - the main features of optimisation, genetic algorithms, fuzzy logic, and expert systems
  - the basic principle, algorithm, and applications of optimisation, genetic algorithms, fuzzy logic, and expert systems
  - the advantages and limitations of intelligent techniques

Introduction

- While this unit has focused on neural networks and data mining, there are many other intelligent techniques that are gaining popularity.
- We will briefly cover the main features of optimisation, genetic algorithms, fuzzy logic, and expert systems.
  - Basic principle, algorithm, and applications.

Optimisation

- Optimisation is the process of making something better. (“**Science of Better**”)
  - Trying variations of a way of doing something to find a better way of doing it, e.g. the fastest route or the best time of day to drive to university in order to minimise travel time or cost.
- In mathematics, optimisation is the process of adjusting inputs (referred to as decisions) to find the minimum or maximum output (expressed as an objective function).

Why Optimisation Matters?

- Resources are limited and valuable:
  - Oil in the earth
  - Land for waste dumps
  - Time
  - Money
  - Workers
- Decision problem:

  To decide how to best use limited resources to:
  - Maximise profits or
  - Minimise costs

Applications of Optimisation

- Determining Product Mix
  - How many of each product to produce to **maximise profits** or to satisfy demand at the **minimum cost**?
- Manufacturing
  - For a circuit board, what is the drilling order that **minimises total distance** the drill bit must be moved?
- Routing and Logistics
  - What is the **least costly** method of transferring goods from warehouses to stores?
- Financial Planning
  - How much money to put into superannuation to **minimise tax liability**?

Characteristics of Optimisation Problems

- Decisions
  - One or more decisions that must be made
- Constraints
  - Due to limited resources
- Objectives
  - Goal that the decision maker considers when making decisions
  - Maximise or Minimise …

Optimisation Methods

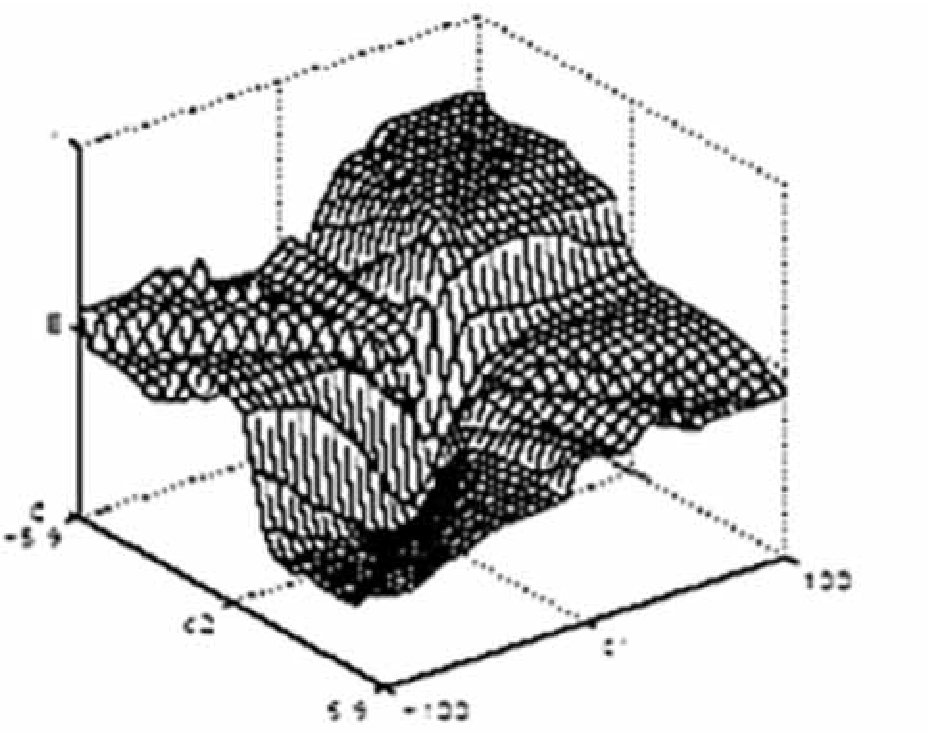
Many methods exist for finding optimal solutions, e.g.

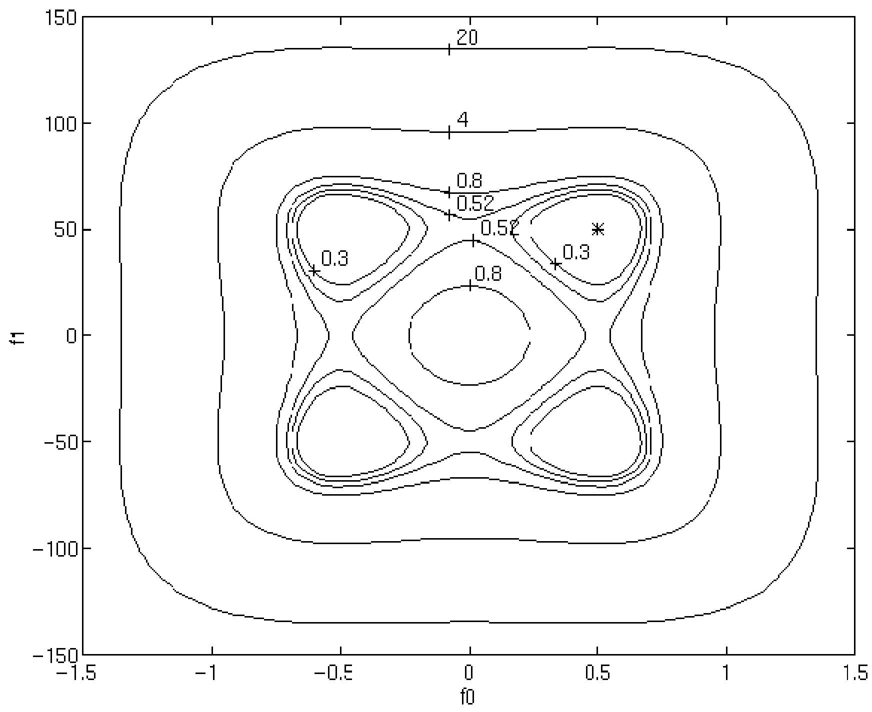
- Generate all possible outputs and look for the one with the lowest error. (Exhaustive enumeration)
- Select a promising sample of outputs and hope that the minimum is close to the global minimum.
- Follow the direction of errors “downhill” to find a minimum (e.g. used in Neural Networks).
- Biological optimisation.

A Review of NN Optimisation

- Supervised Neural Networks seek to minimise the difference between desired (expected) output and actual (observed or predicted) output.
- NN models seek to minimise error.
- "Error surface" example
  - Average *W*, *V* weights form *x*, *y* axes
  - Average error given on *z* axis
  - Adjust *W*, *V* to get the minimum average error (i.e. the best fit of the model)
- Many ways to make the downhill method faster and overcome local minima.
  - e.g. vary initial weights, learning rate, activation function, momentum factor for the backpropogation learning algorithm.

NN Error Surface Examples





- 3-D view
- *x, y* axes - average *W, V* weights
- Z-axis is error at the output neuron
- Delta rule goes “downhill” to find the minimum error

- 2-D view
- *x, y* axes - average *W, V* weights
- Like a contour map: each line joins points with the same error
- Delta rule travels “downhill” to minimum

Biological Optimisation

- Each organism tries to maximise its chances of survival.
- The synthetic theory of natural selection combines 2 strategies:
  - Genetics (micro scale). Traits are inherited from the combination of parent chromosomes.
  - Evolution (macro scale). Children who inherit “good” traits are the most fit to survive (natural selection) – survival of the fittest.

Genetic Algorithms (GAs)

- Mathematical application of genetics and evolution.
- Developed by Professor John Holland at University of Michigan in the 1970's.
  - Based on his studies of natural and artificial systems of adaptation.
- The algorithm has mostly applied to optimisation problems.
  - Aiming at searching the best possible solution to an optimisation from a large set of alternative solutions.

GAs (continued)

- GAs are ideally suited to solving optimisation problems with computational complexity, due to the nature of their search mechanism.
  - The algorithm allows the quality of the solutions to improve naturally over time.
- Many business problems can be regarded as optimisation problems, including classification and prediction.
  - The objective is to minimise the error of the model when classifying or predicting.
  - GAs have gained popularity as a directed knowledge discovery technique for data mining.

GAs (continued)

- GAs attempt to artificially replicate some of the ideas found in genetics.
  - NNs attempts to artificially replicate some of the ideas found in the system of the human brain.
- More specifically, GAs are based on the Darwinian notions of **survival of the fittest** and **natural selection**.
  - According to Darwin (1859), in nature (both plant and animal), the strongest and fittest members of many species tend to survive and reproduce.
  - This nature selection process leads to the genetic material of the fittest individuals being passed on to the next generation, thus ensuring the continuous strengthening of the species.

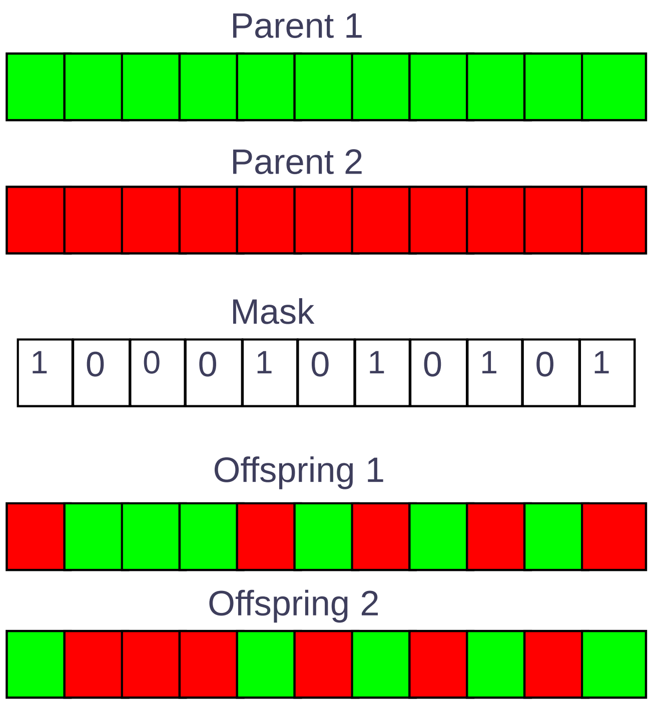
GAs (continued)

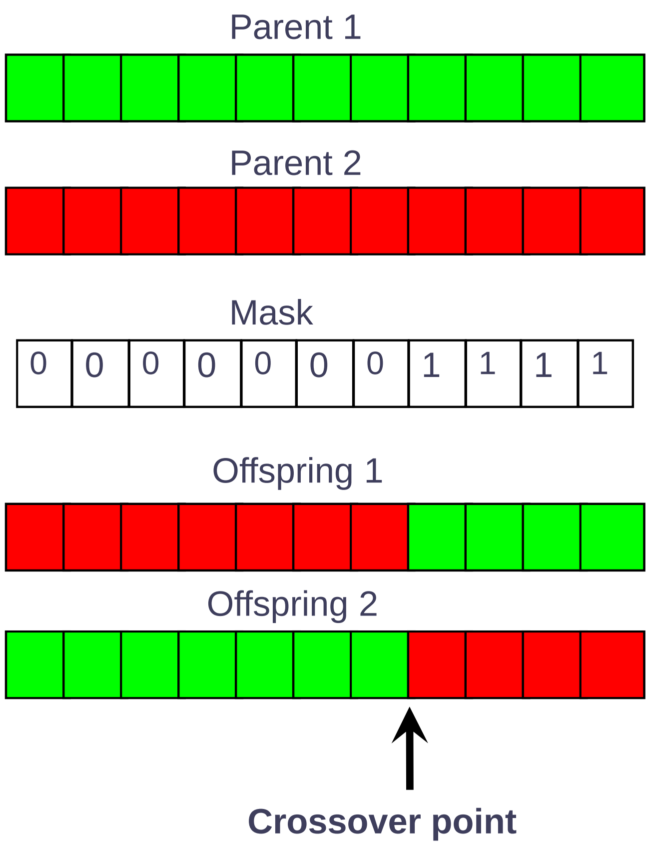
- GAs operates within this similar (yet artificially created) environment.
- An individual is made up of strings of genes, just as humans are made of chromosomes.
- Some genes will indicate that the individual is strong, while others may indicate weakness.
- When two individuals reproduce, the off-spring (an individual of the next generation) inherits a combination of genes from both parents.
- The strength or fitness of individuals will be evaluated for their survival to reproduce.

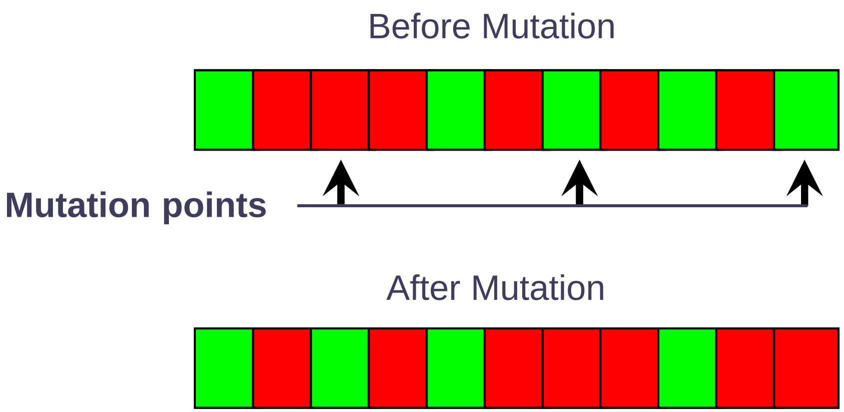
Problem Solving Using GAs

- The population is a group of possible solutions (i.e. the solution space) to a problem.
  - An individual is one solution to the problem.
  - The fitness is calculated for each solution (how well it solves the problem).
  - The most fit solutions survive to reproduce.
  - Reproduction is done by combining solutions (crossover) or by altering existing solutions (mutation).
  - Keep evolving until a sufficiently good solution is achieved.

Crossover and Mutation







A Genetic Algorithm

- Choose a population size *n* (number of solutions).
- Generate *n* random solutions.
- Calculate the fitness of each solution (e.g. the error generated).
- Use the most fit for selective reproduction.
- Calculate the fitness of the new population of solutions.
- Select the best from the offspring and parents to form the next generation.
- Loop to selective reproduction until a good solution is obtained.

Fitness

- Fitness may be measured by how close the solution is to a desired solution.
  - The problem then becomes an optimisation one (minimisation of error).
  - Fitness can be measured by the error that each gene (chromosome) generates.
  - Genes (chromosomes) that give the lowest error in the population survive to reproduce (by crossover and/or mutation).

GA Error Surface

- In searching to minimise error, GAs start at several points (possible solutions) on the error surface and compare the progress at each point.
- The points where the best progress is being made (i.e. with the lowest error) continue being used but are modified slightly (through crossover and mutation).

GAs vs. NNs

- GAs are an optimisation technique but different from the steepest descent approach (used by NNs).
- GAs have many initial starting points, while NNs use one set of initial weights.
- NN optimisation goes gradually downhill. GA crossovers and mutations can “jump” to other solutions (in order to avoid local minima).
- GAs simultaneously search many solutions – NN moves gradually to a solution.
- GAs can provide a list of solutions, not just one, so GAs are potentially more powerful as a search technique.

An Example

- Consider an inventory application:
  - What is the optimal stock level of 7 products to maximise profitability
  - Too much in stock => high storage costs
  - Insufficient stock => missed sales
- One random solution (a stock level of 7 products) might be:

1	97	15	5	27	56	43

Assuming this solution produces a profitability of 10, denoted as P(X) = 10.

An Example (continued)

Solution 1	1	97	15	5	27	56	43	P(X) = 10 (fitness)
Solution 2	105	31	52	35	9	21	16	P(X) = -2

Crossover point

Offspring 1	1	97	15	35	9	21	16	P(X)=-50 really bad
Offspring 2	105	31	52	5	27	56	43	P(X)=96 really good

If this gene was mutated from 5 to say 16, then the fitness of Offspring 2 might further increase.

Another GA

- Suppose we have a population of individuals (X) and the fitness of each X can be measured as F(X).
- The following algorithm will gradually increase the average fitness F(X) across the population of individuals.
- Each generation of individuals will reproduce to produce the next generation, and then die off, with the offspring becoming parents, and reproducing.

Another GA (continued)

- STEP 1:
  - Choose the population size P, and the number of generations G.
- STEP 2:
  - Calculate F(X) for all initial P individuals.
- STEP 3:
  - Select the best 50% of individuals for reproduction.
- STEP 4:
  - Use random crossover until P offspring created.
  - Use random mutation on some offspring.
- STEP 5:
  - Parents die, and offspring become the next generation
- STEP 6:
  - Repeat from STEP 2 for G generations

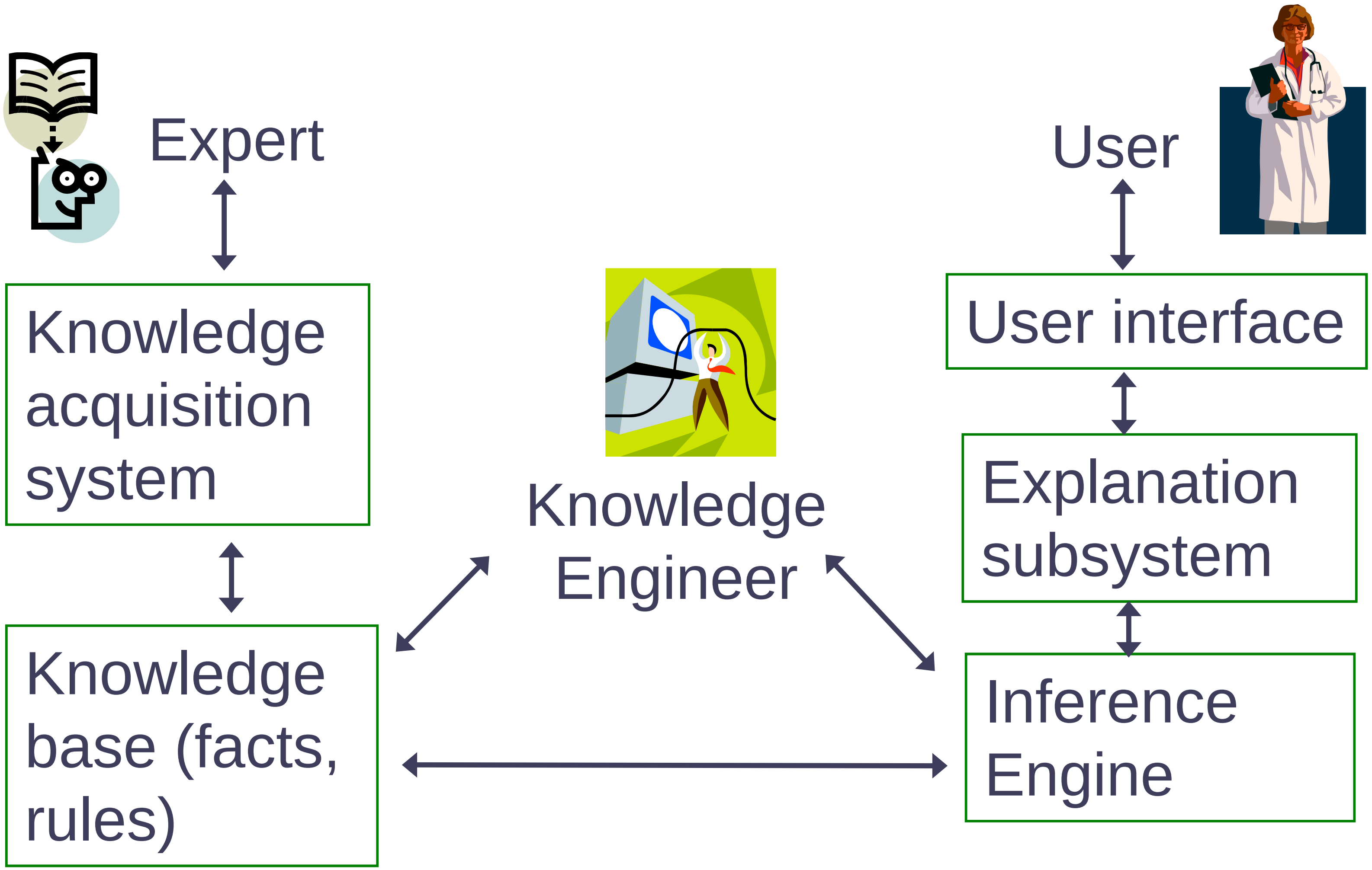
Expert Systems

- Basic principle
  - Instead of trying to mimic human intelligence or thinking/learning patterns, expert systems just get the experts to tell us what they know.
  - This is a finite set of knowledge.
  - Expert systems merely reproduce the answers of an expert.
  - Expert systems do not generalise their understanding to different situations.

Expert Systems (continued)

- Knowledge is usually *elicited* though interview or observation.
- This knowledge is then represented as *Production Rules.*
  - As an example: IF condition1 AND condition2 THEN action1 AND action2, etc.
- One part of the expert system stores these rules, while the other part (the shell) is responsible for reasoning and communication with the user.
  - The Shell part can be purchased, and the rules need added e.g. http://www.expertise2go.com/webesie/

Expert System Structure



Applications of Expert Systems in Business

- Expert Systems grew in popularity during the 1970’s and 1980’s.
  - DENDRAL
  - MYCIN
  - PROSPECTOR
  - INTERNIST
- This is mostly because people were disillusioned with neural networks after Minsky and Papert’s work about limitations of single-layered perceptrons in 1969.
- Also many other human tasks that can be easily replaced.

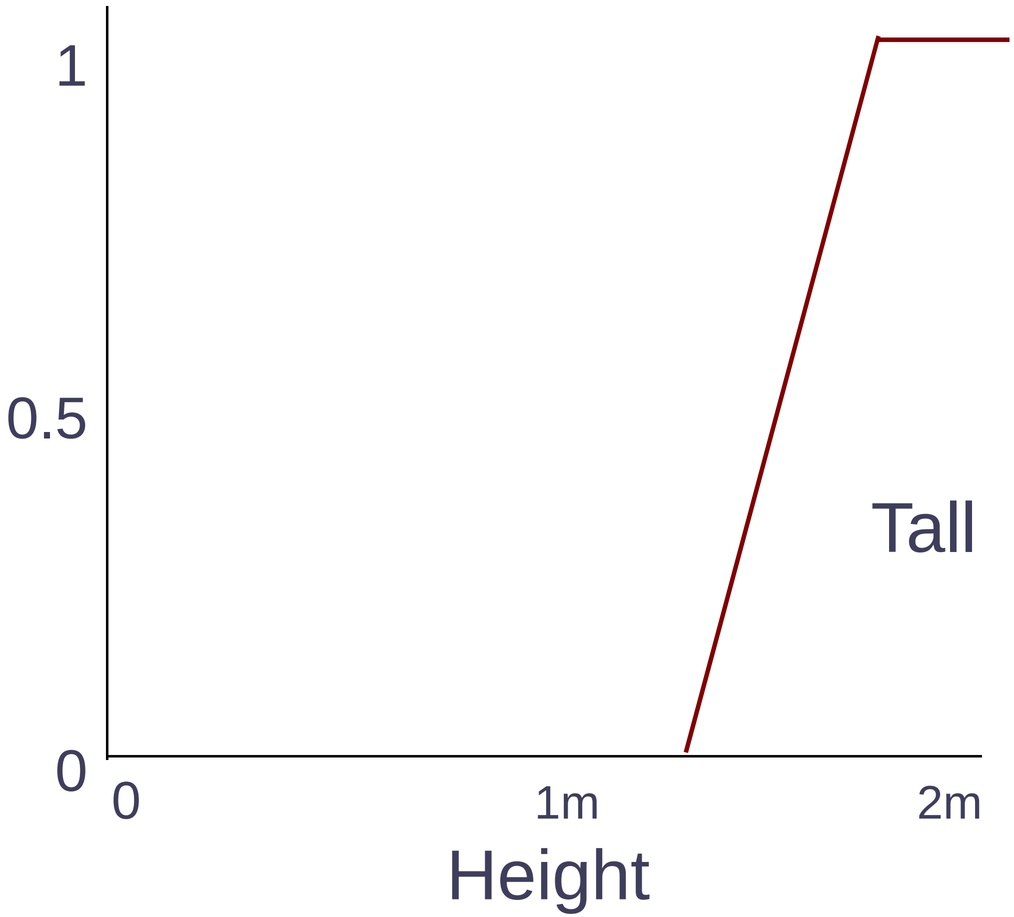
Fuzzy Logic

- Basic principle
  - How can we expect computers to make decisions like humans, when we only allow them to operate using Boolean (true or false) logic?
  - We may say a person's income is high if they earn > $80k, but might say that a person earning $79K also has quite a high income.
  - We may describe someone of 2m height as "tall", but also describe someone of 1.9m height as tall.

Fuzzy Logic (continued)

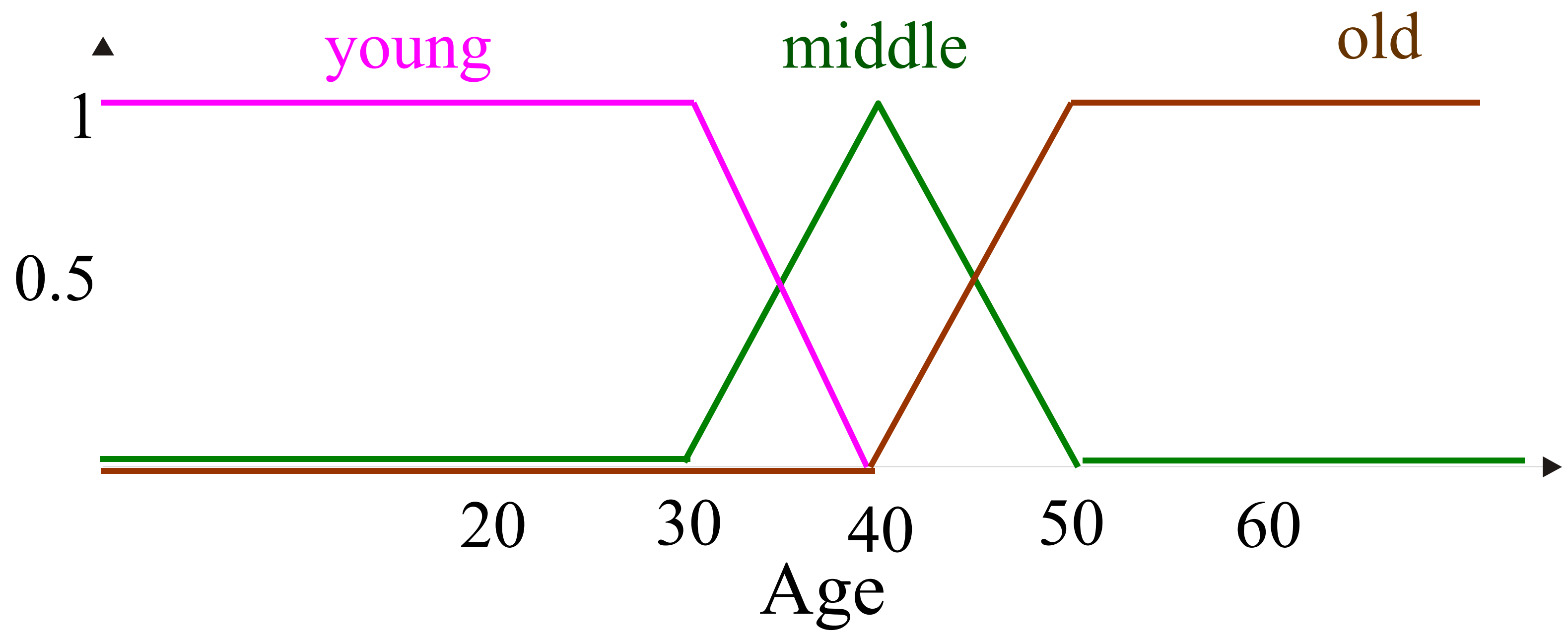
- Fuzzy logic is an attempt to create a mathematical vocabulary for expressing uncertainty and vagueness.
- It allows us to say a variable belongs “a bit” to class A and “a bit more” to class B. (“a matter of degree”).
- This is done through the construction of membership functions.
  - For a given (linguistic) variable (i.e. age or income) we have to decide possible classes or states (e.g. low, middle, high) and then decide how a given value (e.g. age = 45) can best be described.

A Membership Function



“Tall” is a state of the linguistic variable “Height”.

Possible Membership Function for Age



- We might decide that the states of a person's age can be (young, middle, old).
- The membership to (young, middle, old) is a matter of degree
  - e.g. an age of 45 might be considered 50% middle and 50% old i.e. (0, 0.5, 0.5)
- An age of 25 might be considered clearly young (1.0, 0.0, 0.0).
- An age of 40 might be clearly middle aged (0.0, 1.0, 0.0).

Fuzzy Rules

- Once membership functions have been created for all linguistic variables, fuzzy rules can be formed.
  - Similar to rule formation by experts' knowledge.
  - Rules are subjective, as are the membership functions.
- Example:
  - IF age = young AND income = low THEN risk = high
  - Input variables, variable state, decision variable
- Unlike expert system rules, the fuzzy rules are not "switches" but "fuzzy" boundaries.

Fuzzy Rules (continued)

- A set of fuzzy rules can be formed with linguistic variables characterised by membership functions, which are linked using fuzzy AND, OR operators.

- **AND** is represented by:

  The minimum of the memberships, i.e.

  Min{membership value of A, membership value of B}

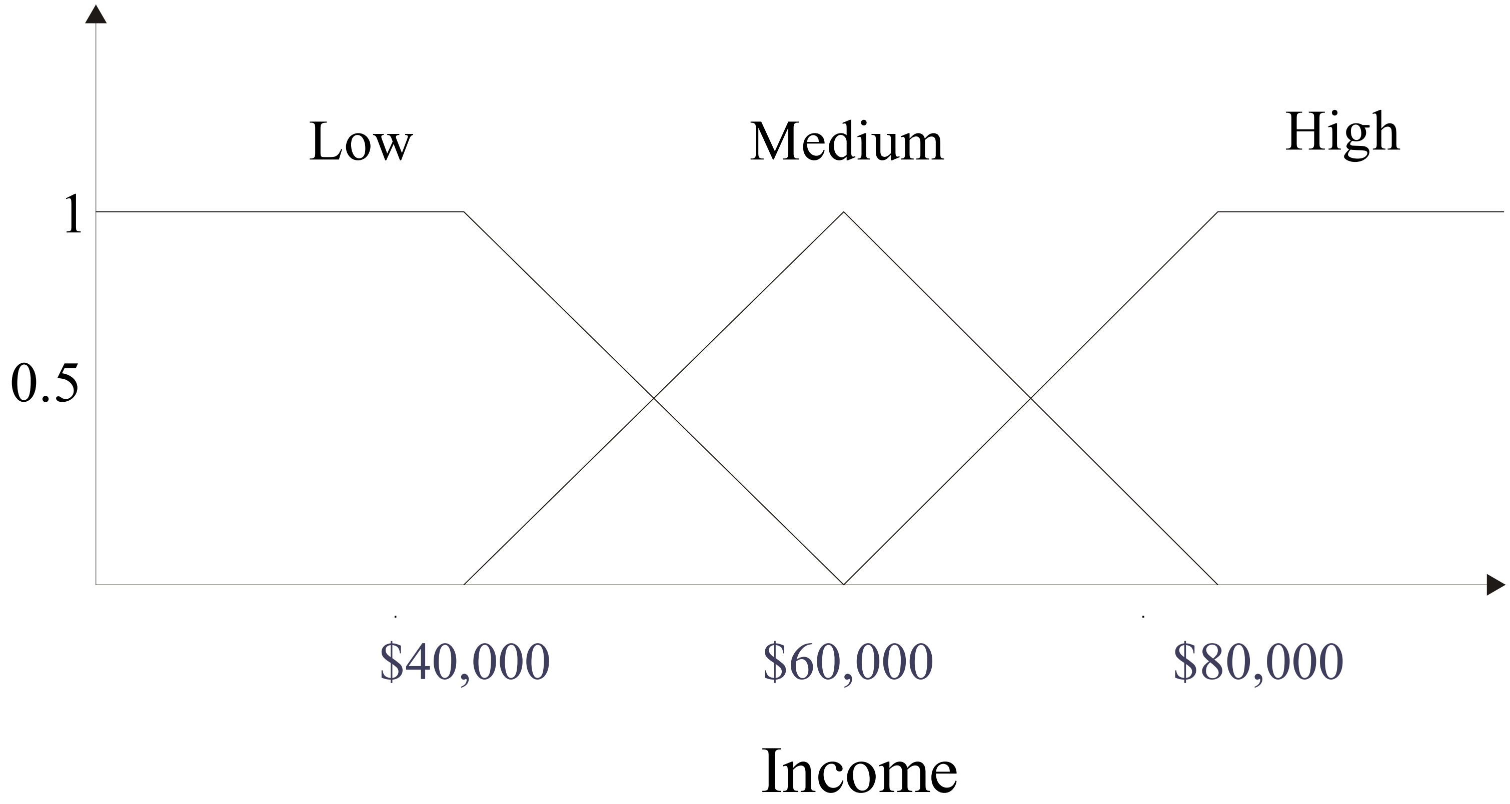
- **OR** is represented by:

  The maximum of the memberships, i.e.

  Max{membership value of A, membership value of B}

Example

- Suppose we are trying to use age and income only to determine credit risk.
  - The age membership functions are given earlier.
  - The income membership functions might be:



Example – Decision Beliefs

- We need to define a set of rules based on the fuzzy variables (i.e. age and income) constructed for indicating credit risk level (e.g. low, medium, high).
- An example of commonly held decision beliefs is as follows:

Age	Income	Risk
Young	High	High
Young	Medium	Medium
Young	Low	High
Middle	High	Low
Middle	Medium	Low
Middle	Low	Medium
Old	High	Low
Old	Medium	Medium
Old	Low	High

Example - Fuzzy Rules

- The decision beliefs can be combined into a set of fuzzy rules for determining credit risk as follows :
  - If (age=young AND income=high) OR (age=young AND income=low) OR (age=old AND income=low) THEN risk=high;
  - If (age=young AND income=medium) OR (age=middle AND income=low) OR (age=old AND income=medium) THEN risk=medium;
  - If (age=middle AND income=high) OR (age=middle AND income=medium) OR (age=old AND income=high) THEN risk=low.

Example – A Case

- Determine credit risk for a person of 25 years old who earns $70,000/year.

- From the membership functions a person of age of 25 with $70k income:
  – Age: (young=1.0, middle=0.0, old=0.0)
  – Income: (low=0.0, medium=0.5, high=0.5)

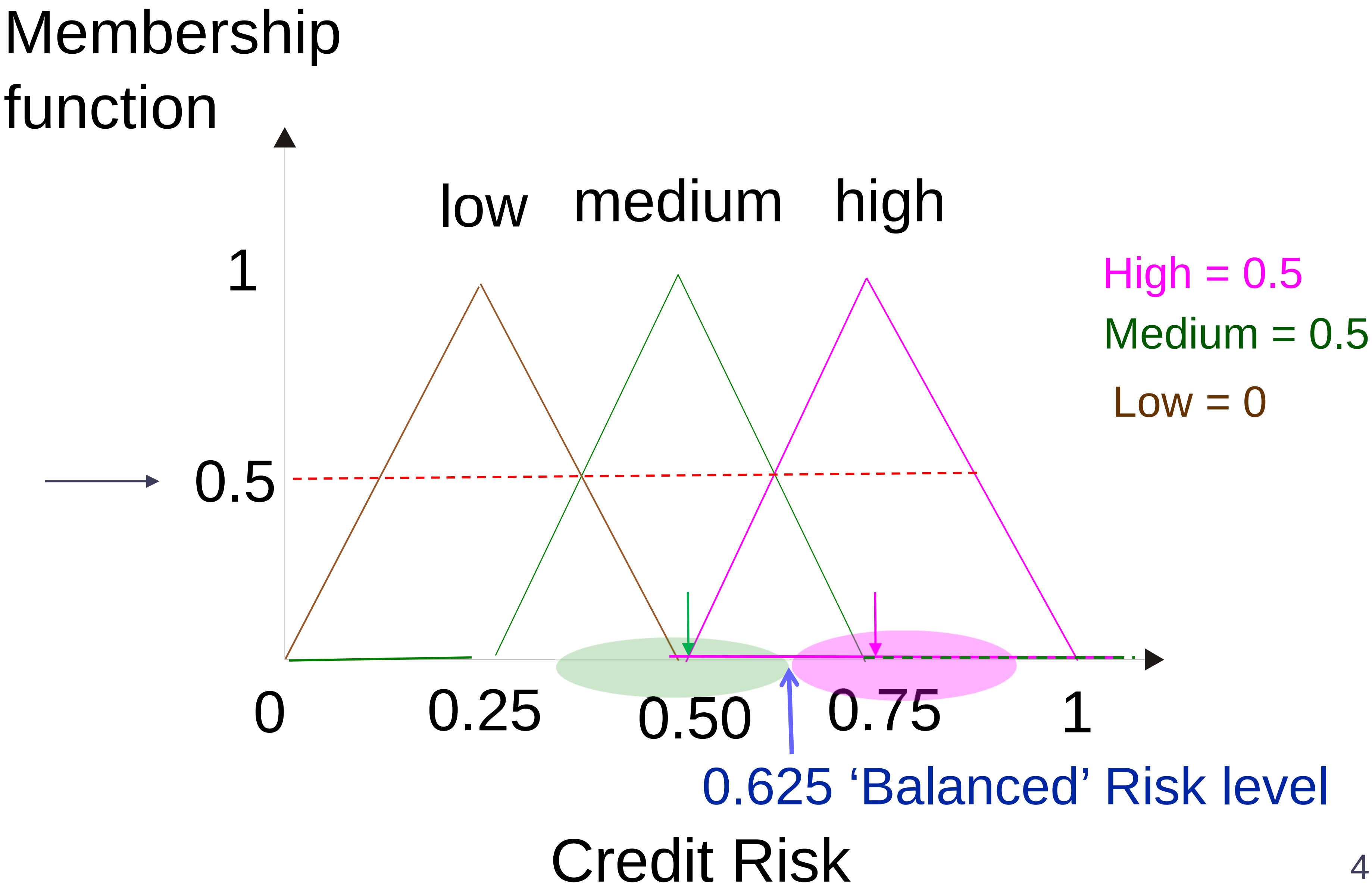
- Substitute into the rules:
  – If (age=young AND income=high) OR (age=young AND income=low) OR (age=old AND income=low) THEN risk=high;
  – So the high risk value = max{min(1, 0.5), min(1,0), min(0,0)} = max{0.5, 0, 0} = 0.5

Example – A Case (continued)

- Similar values can be produced for the medium and low credit risk rules:
  - Medium risk value = 0.5
  - Low risk value = 0.0
  - Final risk: (High. Medium, Low) = (0.5, 0.5, 0.0)

- Ideally we would be able to convert this to a “crisp” value (i.e. to defuzzify the fuzzy output of the fuzzy rules into a scalar, non-fuzzy value), so we can compare different people against established standards.

Defuzzify Credit Risk (0.5, 0.5, 0)



Credit Risk Cases

- The person has credit risk (0.5, 0.5, 0) for the three states of high, medium and low.
- Looking at the membership function for risk this puts them at 0.625 (the midway point).
- If you repeat this example for a 35 year old person on $70k, you find their credit risk is 0.5 (lower).
- So it’s possible to decide which of the 2 customers is a better credit risk (young/medium-high income vs. middle-age/medium-high income).

A Fuzzy Logic Algorithm

- Step 1: Determine appropriate states for linguistic variables and the corresponding membership functions.
- Step 2: Generate fuzzy values for each of the input variables (fuzzification) .
- Step 3: Produce a set of IF-THEN rules to calculate fuzzy outputs.
- Step 4: Map the fuzzy inputs (from Step 2) to the rules to calculate fuzzy outputs (inferencing).
- Step 5: Defuzzify the output to produce a crisp value.

Business Applications

- Fuzzy logic used most heavily in fuzzy controllers for washing machines, air conditioners, vacuum cleaners, windscreen wipers, etc.
- Good for areas like fraud detection, where there is no clear definition of unusual behaviour.
- Also in stock market prediction, using imprecise descriptions of subjective variables.
  - e.g. economic environment, political stability.
  - These are difficult to quantify, but may be useful in rules.

Neural Networks

- Good learning ability, generalisation, and flexibility.
- Poor explanation (no rules are produced that explain why a decision was made).
  - Just “because the neural network said so”.
- Some problems (local minima, slow convergence) because of the steepest descent nature of the learning.

Genetic Algorithms

- Quite good in all areas.
- Good for certain hard problems.
- Can find sets of minima.
- Heavy processing overhead (checks many possible solutions simultaneously).
- Difficult to encode genes to represent a problem
  - Problems with finding effective ways of representing how the genes correspond to a solution.

Expert Systems

- No learning.
- Simply applying a set of rules determined by experts.
- Have proven very successful in certain applications.
- What if these rules are wrong?
  - Not very flexible.
- But explainability is good because it is rule based.

Fuzzy Logic

- No learning.
- A flexible way of mathematically describing subjective or vague rules.
- Computing with words rather than numbers.
- Can be explained because it is rule based rather than complex calculations.

Advantages and Limitations

Technology	Learning	Flexibility	Adaptation	Explanation	Discovery
NN	*****	*****	*****	*	**
GA	*****	****	****	***	*****
Fuzzy	*	*****	*	****	**
Expert	*	*	*	*****	*

- **“No free lunch” theorem** states that, on average, the performance of all search algorithms over all problems is equal. Wolpert (1997)
  - i.e. some algorithms are suited to certain problems, and none are suited to all problems.

Hybridisation of Intelligent Methods

- Each of the intelligent techniques discussed so far has advantages and limitations.
- There is a trend towards hybrid intelligent systems which incorporate

  neural networks, genetic algorithms (evolutionary computation), fuzzy systems, expert systems, and many others.
- These hybrid systems are more likely to be able to solve intractable problems than each of the techniques individually.

Hybrid Systems

- Complex problems (e.g. financial planning) have many sub-tasks.
- Diverse intelligent methods may be used for each of the sub-tasks.
  - Neural networks, GAs, fuzzy logic, expert systems.
- Design of hybrid systems is difficult because of the complex interaction of the different methods.

Hybridisation Schemes

- There are 3 main ways to hybridise or combine intelligent techniques
  - Function-replacing hybrids
  - Intercommunicating hybrids
  - Polymorphic hybrids
- Each of these approaches aims to overcome the limitations of a single method by taking the advantages of several methods.
- Sometimes called the fusion of soft computing (NN, GA, fuzzy logic) and hard computing (expert systems).

Function-Replacing Hybrids

- One part of a method is replaced by another intelligent method.
  - Example: you can use a genetic algorithm instead of the backpropagation learning algorithm to find the optimal set of weights for a neural network.
    - As a result: NN(GA)
  - Example: you can use a neural network to find fuzzy rules from fuzzy data (rather than get experts to produce rules).
    - As a result: Fuzzy(NN) or NeuroFuzzy

Intercommunicating Hybrids

- The results of one intelligent method are passed onto another method (outputs of one become the inputs of the other).
- *Agent based* computer design is used to facilitate building of such hybrid systems.
  - *Agents* are encapsulated computer programs.
  - Agents work together to solve a problem that are beyond the individual agent's capabilities.
  - There is no global system control over the agents.

Hybrid Systems - Examples

- Example: use a GA to evolve an optimal architecture for a NN (i.e. number of hidden neurons).
  - As a result: GA + NN
- Example: you can use fuzzy logic to decide the economic and political environment (subjective elements), and then add that to a neural network that considers financial data.
  - As a result: Fuzzy + NN

Polymorphic hybrids

- Different methods within the hybrid are selected in different problem situations.
  - May also be agent based.
  - Fuzzy or NN + expert system for flight controller.

Selected Readings

- Rule based expert systems
  http://www.aaai.org/Resources/Classics/Mycin/mycin.html
- New Tools Help Hospitals Handle Terror Attacks And Other Disasters
  http://informationweek.com/story/showArticle.jhtml?articleID=160900664
- Engelbrecht (2007). Computational Intelligence – An Introduction, 2nd Edition, Wiley.
- Ovaska (2005). Computationally Intelligent Hybrid Systems, IEEE Press.

Week 9 Tutorial

- SOFM using Viscovery SOMine.
- 52 countries with economic data.
- Use SOFM to cluster, and compare economic clusters to those categories from the Wall Street Journal (see the tutorial sheet).
- Experiment with the cluster threshold.
- Appreciate the visual representation of the data in this way.