Pattern Recognition Numerical

n=165	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

What can we learn from this matrix?

- There are two possible predicted classes: "yes" and "no". If we were predicting the presence of a disease, for example, "yes" would mean they have the disease, and "no" would mean they don't have the disease.
- The classifier made a total of 165 predictions (e.g., 165 patients were being tested for the presence of that disease).
- Out of those 165 cases, the classifier predicted "yes" 110 times, and "no" 55 times.
- In reality, 105 patients in the sample have the disease, and 60 patients do not.

Let's now define the most basic terms, which are whole numbers (not rates):

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Now, added these terms to the confusion matrix, and also added the row and column totals:

n=165	**Predicted: NO**	**Predicted: YES**	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

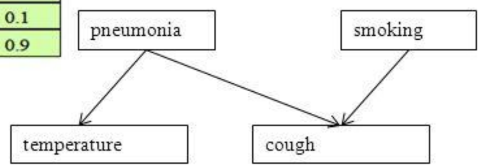
This is a list of rates that are often computed from a confusion matrix for a binary classifier:

- **Accuracy:** Overall, how often is the classifier correct?
  - (TP+TN)/total = (100+50)/165 = 0.91
- **Misclassification Rate:** Overall, how often is it wrong?
  - (FP+FN)/total = (10+5)/165 = 0.09
  - equivalent to 1 minus Accuracy
  - also known as "Error Rate"
- **True Positive Rate:** When it's actually yes, how often does it predict yes?
  - TP/actual yes = 100/105 = 0.95
  - also known as "Sensitivity" or "Recall"
- **False Positive Rate:** When it's actually no, how often does it predict yes?
  - FP/actual no = 10/60 = 0.17
- **True Negative Rate:** When it's actually no, how often does it predict no?
  - TN/actual no = 50/60 = 0.83
  - equivalent to 1 minus False Positive Rate
  - also known as "Specificity"
- **Precision:** When it predicts yes, how often is it correct?
  - TP/predicted yes = 100/110 = 0.91
- **Prevalence:** How often does the yes condition actually occur in our sample?
  - actual yes/total = 105/165 = 0.64

- **Diagnostic: Evidence:** *cough=true.* What is *P*(*pneumonia* | *cough*)?

pneumonia	
true	0.1
false	0.9

smoking	
yes	0.2
no	0.8



	temperature	
pneumonia	yes	no
yes	0.9	0.1
no	0.2	0.8

		cough	
pneumonia	smoking	true	false
true	yes	0.95	0.05
true	no	0.8	0.2
false	yes	0.6	0.4
false	no	0.05	0.95

$$P(pneumonia \mid cough) = \frac{P(cough \mid pneumonia) P(pneumonia)}{P(cough)}$$

$$= \frac{[P(cough \mid pneumonia, smoking) P(smoking) + P(cough \mid pneumonia, \neg smoking) P(\neg smoking)] P(pneumonia)}{P(cough)}$$

$$= \frac{[(.95)(.2) + (.8)(.8)](.1)}{P(cough)} = \frac{.083}{P(cough)}$$

$$= \frac{.083}{P(cough)} = \frac{.083}{.227} = .366$$

Q. While watching a game of Champions League football in a cafe, you observe someone who is clearly supporting Manchester United in the game. Using Bayes Rule Calculate the probability that they ~~are~~ were actually born within 30 miles of Manchester. Assume that:

- The probability that a randomly selected person in a typical local bar environment is born within 30 miles of Manchester is $\frac{1}{20}$
- The chance that a person born within 30 miles of Manchester actually supports Manchester United is 7/10
- The probability that a person not born within 30 miles of Manchester supports Manchester United with probability 1/10

Solution:

Let M: Set of born within 30 miles of Manchester

N: " " Not " " " " " "

S: Set of Supporter of Manchester

Here, given,

$$P(M) = \frac{1}{20}$$

$$P(N) = 1 - \frac{1}{20} = \frac{19}{20}$$

$$P(M|S) = \frac{7}{10}$$

$$P(N|S) = 1/10$$

$$P(S|M) = ?$$

$$P(S|M) = \frac{P(M) \cdot P(M|S)}{P(M) \cdot P(M|S) + P(N) \cdot P(N|S)}$$

$$= \frac{\frac{1}{20} * \frac{7}{10}}{\frac{1}{20} * \frac{7}{10} + \frac{19}{20} * \frac{1}{20}}$$

$$= \frac{7}{26}$$

Unit 8 Swarm Intelligent

Introduction

- Nature has guided us to watch and learn the intelligent mechanism evolved by it, like marching of ant in an army, the waggle dance of honey bee, the nest building of social wasp, birds flocking in high skies, fish schools in deep waters, foraging activities of microorganisms, the construction of termite mound, etc. These activities has puzzeled biologists over the years.
- The last decade has seen an explosion of research in this field variously reffered to as collective intelligence, swarm intelligence and emergent behaviour.
- Many instances of creativity in animals arises from *collective behavior*, not from single individual's actions
- Swarm Intelligence is used for the collective behavior of a group of animals as a single living creature, where collective intelligence emerges via grouping and communication, resulting in more successful foraging for each individual in a group.
- In the recent past, the swarm paradigm has been applied to broader range of studies , opening up new views of therotical biology, economics and philosophy.
- ***Adaptation:*** Swarm Intelligence is a specialization in the field of self-organizing systems
- ***Robustness*:** When a route of ant is blocked, this can be observed that they find other new shortest new route to their destination.
- ***Reliable*:** Agents can be added or removed without compromising the total system due to its distributed nature. This adaption is called Reliable.
- ***Simplicity*:** Single part may break down without impairing the overall system such that complex system are convenient to work
- The desired characteristics emerge from the interaction of the various parts without explicit supervision or central control system which is intelligent behavior
- Knowledge is distributed and becomes apparent in the interaction between the agents and the environment
- *Optimal foraging policy :* Animal search for and obtain nutrients in a way that maximizes their energy intake per unit time spent foraging. The maximization of such a function provides nutrient sources for the animal to survive and additonal time for other important activities. The foraging formulation is only meant to be a model that explains optimal behaviour.

Background of Ant Intelligent System

- In 1991 Marco Dorigo and his colleague proposed ant algorithm as multi agent approach to solve difficult combinatorial optimization problem
- There is currently a lot of ongong activity in the scientific community to extend/apply ant-based algorithms to many different discrete optimizaton problems
- The application and experimental validation of these algorithms are being thoroughly researched owing to their capability to provide an almost global optimal aolution to a given complex problem structure such as local search, image mapping and compression, database search etc.
- Ant algorithms were first tested and validated on the Travelling Salesman Problem
    - Reasons behind choosing TSP are:
        - It is a shortest path problem for which the ant colony metaphor can easily be adopted
        - The main idea is that of having a set of agents, called ants, search in parallel for good solutions and cooperate through the pheromone-mediated indirect method of communication

Importance

- Ant Colony Intelligent System is being used as an intelligent tools to help researchers to solve many problems in different areas of science and technology
- Scientists, now a days, are using real ant colonies to solve many combinatorial optimization problems in different engineering applications
- Traditional, sequential, logic-based digital computing excels in many years
- Features like positive feedback, distributed computation, and constructive greedy heuristics approaches made ACIS successful with tremendous potential

Ant Colony System

- ❑ An artificial ant colony system is a random stochastic population-based heuristic algorithm of agent that simulate the natural behavior of ants, developing mechanisms of cooperation and learning, which enables the exploration of the positive feedback between agents as a search mechanism.
- ❑ **Biological Ant colony System**
    - ❑ Ants perform their task autonomously without central coordination

- When they act as a community they can solve complex problems emerging in their daily lives through mutual cooperation known as swarm intelligence

- Swarm intelligence has four basic ingredients **a**) positive feedback **b**) negative feedback (saturation, exhaustion, competition) **c**)amplification of fluctuation (random walk) **d**) mutual interaction

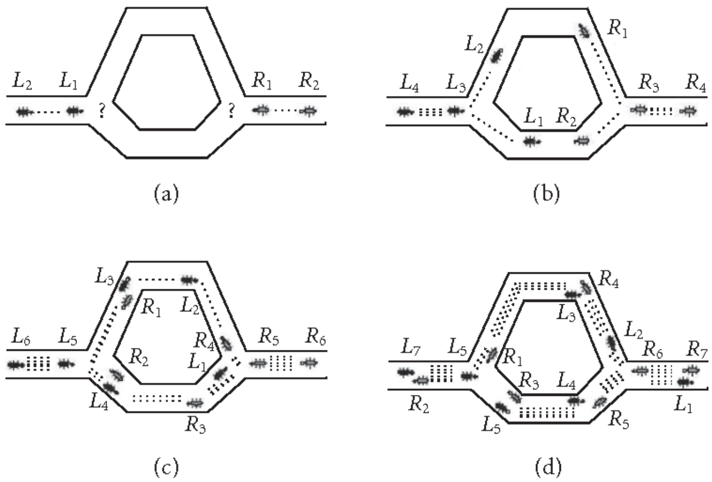


Fig: Foraging behaviour of ants moving from their nest (origin) to the fooed source (destination), taking the shortest possible route through pheromone mediation [stage (a) to stage (d)]

- Ant can smell pheromones; while choosing their path, they tend to choose tha paths marked by strong pheromone concentrations. This trail allows ants to find their way back to the food.

Artificial Ant colony System

- A colony of cooperating individuals, an artificial pheromone trail, a sequence of local moves for finding the shortest path, a stochastic decision policy using local information

- Artificial ant lives in a discrete world and their moves consists of transitions from discrete state to discrete state

- Artificial ants have an internal states

- Artificial ants deposits a particular amount of pheromone, which is the function of the quality of the solution found

Development of Ant Colony System

- Ant System was the first example of an Ant Colony Optimization which consists of three algorithms which was proposed by Dorigo in 1992
    - Ant Cycle
    - Ant Density
    - Ant quality
- In ant density and ant quality, ants can update pheromone trail directly after a move from one node to an adjacent node.
- In ant cycle update was carried out only after all the ants had constructed their tours and amount of pheromone deposited by each ant was set of function denoting the tour quality

<u>Application of Ant Colony Intelligence</u>

- The application of Ant Colony Intelligence is distinguished in two areas:
- ***Static Combinatorial Optimization Problem and Dynamic Combinatorial Optimization Problem***
- *Static Combinatorial Optimization Problem:*
    - characteristics of problem are defined and do not chance till the problem is solved
    - Ex: TSP, Graph coloring, vehicle routing etc
- *Dynamic Combinatorial Optimization Problem*
    - Some values are set by the dynamics of an underlying system
    - Problem changes on the fly so runtime and optimization algorithm must be capable of adapting on the fly to changing environment
    - Ex: Communication Networks

<u>Working of Ant Colony System</u>

- Focuses on following two rules
    - Specifying how ants construct or modify a solution for the problem in hand
    - Updating pheromone trail
- Incorporates two basic activities:
    - ***Probabilistic transition rule:***

- In a simple ACO algorithm, the main taxk of each artificial ant, similar to their natural counterparts, is to find a shortest path between a pair of nodes on a geaph on which the problem representation is suitably mapped.
- Let G=(N,A) be a connected graphwith n=|N| nodes.
- Simple ant colony optimization algorithm can be usedto find the solution to a shortest path problem defined on the graph G, where a solution is a path on the graph on the graph connection a source node to a destination node.
- Path length given by arc (i,j), variable $t_{ij.}$
- Amount of pheromone t_0 is assigned to all the arcs.
- The decision rules of an ant k located in node i use the pheromone trails t_{ij} to compute the probability with which it should choose node $j \in N_i$ as the next node to move, where N_i is the set of one-step neighbous of node i:

$$P_{ij}^{k} = \begin{cases} t_{ij} \Big/ \sum_{j \in N_i} t_{ij} & \text{if } j \in N_i \\ 0 & \text{if } j \not\in N_i \end{cases}$$

❑ Pheromone Updating

- While building a solution , ants deposits pheromone information on the arcs they use.
- They deposit a constant amount Δt of pheromone.
- Consider an ant that at time T moves from node i to j. It will change the pheromone value t_{ij} as follows:
  $T_{ij}(T) \longleftarrow t_{ij}(T) + \Delta t$
- Using this rule, which simulates real ants' pheromone deposits on arc (i,j), an ant using the arc connecting nose i to node j increases the probability that other ants will use the same arc in the future.

Types of Ant Colony Models

1.) Ant colony optimization meta-heuristic model for discrete optimization problems
2.) Ant system model for the travelling salesman problem
3.) Ant-density
4.) Ant-quantity
5.) Ant-Cycle

6.) AS-JSP (for the jo shop scheduling problem)
7.) Ant-Q
8.) Ant colony system-3-opt
9.) AS-QAP(for the quadratic assignment problem)
10.) AS ranking model
11.) Ant net
12.) Ant-based control
13.) Ant net-FA
14.) Ant net-FS
15.) Ant miner for rule discovery in databases.

ANT COLONY ALGORITHM

Ants Foraging Behavior and Optimization

Ant colony optimization (ACO) algorithms are inspired by the foraging behavior of ants in the nature. In nature, some species of ants in searching for food will leave chemicals that can be smelled by

others on the route, called pheromones. By releasing pheromones, ants can mark the route they have walked, providing clues for other ants foraging for food. As time and the number of foraging ants

increase, the concentration of pheromone in the environment will change, based on which ants can gradually find the shortest route between their nest and the food. The most famous monitoring experiment on ants' foraging behavior is the "double bridge experiment" conducted by Deneubourg and his colleagues. They used a double bridge to connect an ants nest and the food source, studying the release of ants pheromone and its influence on ants' foraging behavior. Assume two routes, a long one and a short one, connecting the nest to food. At first, ants chose both routes, but, after a while, all the ants chose the shorter one. In this experiment, at the initial stage, none of the two routes have pheromone, so ants had no reference for choosing, so there's no bias, and ants can only choose at random, so the probability of choosing one of the two routes is equal. However, the ants choosing the shorter route would arrive at the food source earlier, so they can get back earlier. They leave the pheromone on the shorter route next time, meaning that there is much more pheromone on the shorter route than on the longer one during the same period. Thus, as time passes by, accumulative concentrations of pheromone become greater and greater, compared with the shorter one, and the ants followed tend to choose the shorter route. Inspired by this experiment, we can optimize the characteristics of real ants to make ants accurately remember the walked route that can release the pheromone according to certain values. The earliest ant colony algorithm is formed by using ants' foraging behavior in order to solve problems. Artificial Ants and Real Ants Artificial ants and real ants are adopted to distinguish the ant colony algorithm from ants in nature, with comparison between the two. Common points shared by artificial ant and real ant. As the ants in the ant colony algorithm are abstracted from the real ants in nature, the two share many similarities: First, the two groups have communication mechanism. Whether in a real ant colony system or an artificial ant colony system, ants can release pheromones to change their environment, and communicate indirectly with other ants in the ant colony. Second, they complete the same task through collaboration with other ants in the ant colony. Whether it is a real ant or an artificial one, the task is assigned to different ants in solving problems, and all the ants accomplish the same task through mutual cooperation. Third, they all make use of the existing information to choose the route. Both real and artificial ants choose the route or node to be walked through according to the information of the environment they are in. The differences between artificial ants and real ants. In order to meet the demands of the algorithm, many characteristics that real ants don' t have are added during the construction of artificial ants, mainly including:

1. Different from real ants, the artificial ants are in an environment that is a set of discrete state while solving problem, their movement is from one state to another one.

2. Real ants themselves can't record the route walked. But while setting artificial ants, make them capable of recording the routes they have walked through.

3. According to different problems, artificial ants can choose the updating method of pheromone, rather than update the pheromone each time they have moved a step, like real ants do.

4. According to the needs of the problem, artificial ants can add some functions which real ants don' t have, such as adjusting the route selection plans dynamically, and overall updating, and so on.

Basic Ant Colony Algorithm

The ant system (AS) is the earliest algorithm model proposed in the ACO algorithm. In the AS, the ant k on the node i calculates the probability of reaching each node, according to the probability selection formula, and based on its result P_{ij} selects the next node j. Assume $tabu_k$ as the collection of all the nodes that ant k has visited currently, also known as the tab_u list of the ant, and add each node the ant has visited into the collection $tabu_k$.

$$P_{ij} = \frac{\tau_{ij}^{\alpha} \eta_{ij} \beta}{\sum_{h \in N_k} \tau_{ih}^{\alpha} \eta_{ih}^{\beta}}$$

Where in, $N_k = \{C - tabu_k\}$ is the collection of all the nodes that ant k can visit currently, and when all the nodes in C are added into $tabu_k$, it means that all the nodes have been visited by the ant, with the ant coming back to the initial node. τ_{ij} is the amount of information along the route (i, j). η_{ij} is the heuristic function, with its expression in formula, the reciprocal of the distance d_{ij} between the two neighboring cities i, j. α is the information heuristic factor, and β is the expected heuristic factor, with α ,β, standing for the impact level on the algorithm of pheromone and heuristic information, respectively.

$$\eta_{ij}(t) = \frac{1}{d_{ij}}$$

In the AS algorithm, when the ant finished one adventure, it will update the pheromone along the walked path using such a method: make all the ants passing through the route release pheromone, and determine the value of pheromone to be updated, according to the following expression.

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k}$$

Wherein, ρ is the coefficient of pheromone evaporation, a value ranging from [0,1], used to show the evaporation degree of pheromone as time goes by; $\Delta\tau_{ij}^{k}$, is the pheromone value released by the number *k* ant on the route (*i*, *j*) in this iteration process, whose value is determined according to expression .

$$\Delta\tau_{ij}^{k} = \begin{cases} F(k) \text{ if } (i,j) \text{ is an edge of ant } k\text{'s solutions} \\ 0 \quad\quad \text{if } (i,j) \text{ isn't an edge of ant } k\text{'s solutions} \end{cases}$$

Wherein, *F*(*k*) is the pheromone value left by ant *k* on the walked path (*i*, *j*), its size is often proportional to the reciprocal of the route's length solved by ant *k*, and multiply the reciprocal of the route's length by a constant *Q*, and *Q* is used to express the pheromone intensity. So, the shorter the route to be solved is, the bigger the value of *F*(*k*) will be, and the more pheromones are released on this route by ants.

PARTICLE SWARM OPTIMIZATION

Particle swarm optimization (PSO), originated from the analysis of behavior of birds catching food, was put forward by American scholars in the early 1990s. American scholars Kennedy and Eberhart found, during their analysis, that the flying birds often scattered, concentrated, or changed directions in an instant, adjusting their flight – fact, which is usually unexpected. After summarizing the rules, they found that the flying pace of the whole flock of birds would generally keep consistent, and a proper distance was maintained between each individual bird. Through analyzing constantly the behavior of other social animals, such as birds, fishes, ants, and so on, they concluded that, in the behavior rules of social animals, there has been an invisible information sharing platform for those seemingly unstructured and dispersed biological groups. Inspired by this, scholars simulated the behavior of birds constantly, and proposed the concept of particle swarm optimization. Particle swarm optimization has become a better-developed optimization algorithm, in recent years. It searches the optimal solution through continuous iteration, and it finally employs the size of the value of objective function, or the function to be optimized (also known as the fitness function in the particle swarm), in order to evaluate the quality of the solution. For the convenience of the study, birds are considered as particles of life without mass and volume in the algorithm. The algorithm initializes the position of each particle into the solution of problems to be optimized. In the movement process of the particle swarm, information is conveyed between each individual influencing the others, and a particle's moving state is influenced by the speed and direction of its colleagues, and of the whole particle swarm, so that each particle adjusts its own speed and direction according to the historical optimal positions of itself and its colleagues, and keeps flying and searching for the optimal position – the optimal solution. In the process of flying, particles update their position and direction according to their and external information; this has proved that the particle has the memory function, and particles with good positions and directions have the tendency to approach the optimal solution. As such, optimization is done through competition and cooperation between particles.

The procedure of basic particle swarm optimization is as follows:

Step 1. Initialize the parameters: initialize the position and speed of the particle to random numbers in the D-dimensional search space.

Step 2. Evaluate the particle's position: use a fitness function to evaluate each particle's position.

Step 3. Make a comparison between: (1) compare the fitness value of step 2 with the particle's personal best value *pbest*, and make the best value become the newest *pbest*; (2) compare the particle's fitness value with the global best value *gbest*, and the best one becomes *gbest*.

Step 4. Update the particle: according to the formula and, update the particle's speed and position

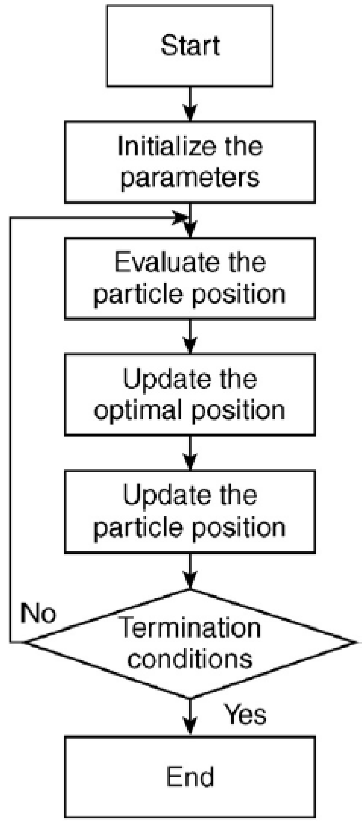


Figure 2.5 The procedure of basic particle swarm optimization.