1

Swarm Intelligence in Data Mining

Crina Grosan[1], Ajith Abraham[2] and Monica Chis[3]

[1] Department of Computer Science
Babeş-Bolyai University, Cluj-Napoca, 3400, Romania
`cgrosan@cs.ubbcluj.ro`
[2] IITA Professorship Program,
School of Computer Science and Engineering
Chung-Ang University, Seoul 156-756, Korea
`ajith.abraham@ieee.org`
[3] Avram Iancu University, Ilie Macelaru 1, 3400
Cluj-Napoca, Romania
`mchis@artelecom.net`

Summary

This chapter presents the biological motivation and some of the theoretical concepts of swarm intelligence with an emphasis on particle swarm optimization and ant colony optimization algorithms. The basic data mining terminologies are explained and linked with some of the past and ongoing works using swarm intelligence techniques.

1.1 Biological Collective Behavior

Swarm behavior can be seen in bird flocks, fish schools, as well as in insects like mosquitoes and midges. Many animal groups such as fish schools and bird flocks clearly display structural order, with the behavior of the organisms so integrated that even though they may change shape and direction, they appear to move as a single coherent entity [11]. The main principles of the collective behavior as presented in Figure 1.1 are:

- *Homogeneity*: every bird in flock has the same behavior model. The flock moves without a leader, even though temporary leaders seem to appear.
- *Locality*: the motion of each bird is only influenced by its nearest flock mates. Vision is considered to be the most important senses for flock organization.
- *Collision Avoidance*: avoid with nearby flock mates.
- *Velocity Matching* : attempt to match velocity with nearby flock mates.
- *Flock Centering*: attempt to stay close to nearby flock mates
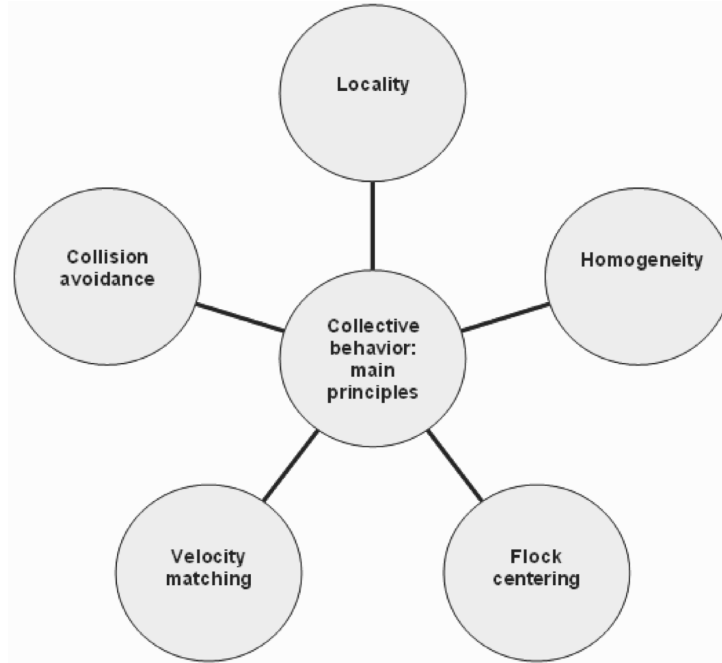
Fig. 1.1. The main principles of collective behavior.

Individuals attempt to maintain a minimum distance between themselves and others at all times. This rule has the highest priority and corresponds to a frequently observed behavior of animals in nature [36]. If individuals are not performing an avoidance manoeuvre, they tend to be attracted towards other individuals (to avoid being isolated) and to align themselves with neighbors [50], [51].

Couzin et al. [11] identified four collective dynamical behaviors as illustrated in Figure 1.1:

- *Swarm* : an aggregate with cohesion, but a low level of polarization (parallel alignment) among members
- *Torus*: individuals perpetually rotate around an empty core (milling). The direction of rotation is random.
- *Dynamic parallel group*: the individuals are polarized and move as a coherent group, but individuals can move throughout the group and density and group form can fluctuate [42], [50].
- *Highly parallel group*: much more static in terms of exchange of spatial positions within the group than the dynamic parallel group and the variation in density and form is minimal.

As mentioned in [22], at a high-level, a swarm can be viewed as a group of agents cooperating to achieve some purposeful behavior and achieve some goal (see Figure 1.3). This collective intelligence seems to emerge from what are often large
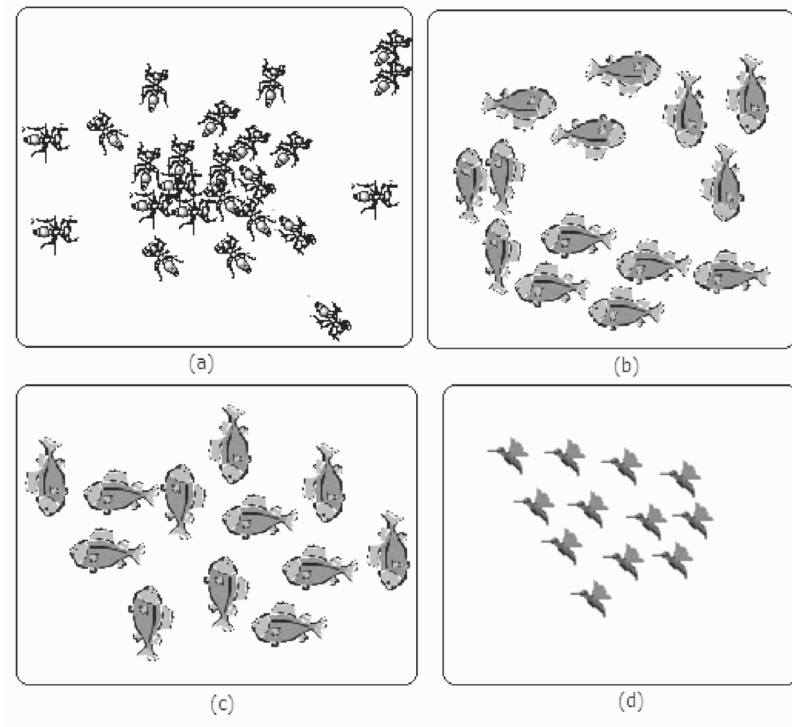
Fig. 1.2. Several models of collective behavior: (a) swarm (b) torus (c) dynamic parallel group and (d) highly parallel group.

groups of relatively simple agents. The agents use simple local rules to govern their actions and via the interactions of the entire group, the swarm achieves its objectives. A type of self-organization emerges from the collection of actions of the group.

An autonomous agent is a subsystem that interacts with its environment, which probably consists of other agents, but acts relatively independently from all other agents [22]. The autonomous agent does not follow commands from a leader, or some global plan [23]. For example, for a bird to participate in a flock, it only adjusts its movements to coordinate with the movements of its flock mates, typically its neighbors that are close to it in the flock. A bird in a flock simply tries to stay close to its neighbors, but avoid collisions with them. Each bird does not take commands from any leader bird since there is no lead bird. Any bird can in the front, center and back of the swarm. Swarm behavior helps birds take advantage of several things including protection from predators (especially for birds in the middle of the flock), and searching for food (essentially each bird is exploiting the eyes of every other bird) [22].
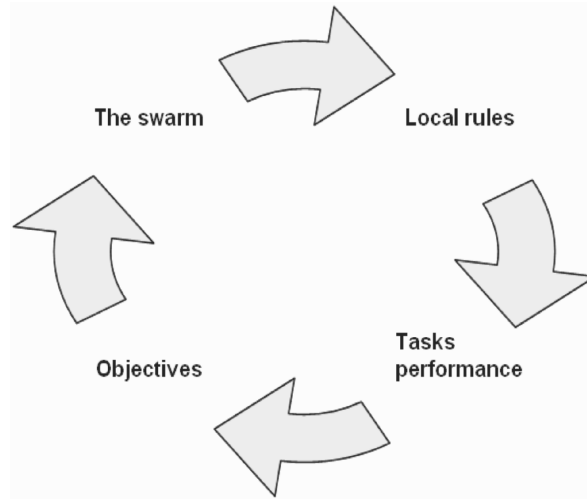
Fig. 1.3. The simple scheme of a swarm.

1.2 Swarms and Artificial Life

Since 1990, several collective behavior (like social insects, bird flocking) inspired algorithms have been proposed. The application areas of these algorithms refer to well studied optimization problems like NP-hard problems (Traveling Salesman Problem, Quadratic Assignment Problem, Graph problems), network routing, clustering, data mining, job scheduling etc.

(PSO) and Ant Colonies Optimization (ACO) are currently the most popular algorithms in the swarm intelligence domain.

1.2.1 Particle Swarm Optimization (PSO)

PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles [26]. Unlike in the other evolutionary computation techniques, each particle in PSO is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviors. Therefore, the particles have the tendency to fly towards the better and better search area over the course of search process. The PSO was first designed to simulate birds seeking food which is defined as a 'cornfield vector' [29], [30], [31], [32], [33].

Assume the following scenario: a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. The birds do not know where the food is. But they know how far the food is and their peers' positions. So what's the best strategy to find the food? An effective strategy is to follow the bird which is nearest to the food.

PSO learns from the scenario and uses it to solve the optimization problems. In PSO, each single solution is like a 'bird' in the search space, which is called 'particle'. All particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. (The particles fly through the problem space by following the particles with the best solutions so far). PSO is initialized with a group of random particles (solutions) and then searches for optima by updating each generation.

Each individual is treated as a volume-less particle (a point) in the D-dimensional search space. The ith particle is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. At each generation, each particle is updated by the following two 'best' values. The first one is the best previous location (the position giving the best fitness value) a particle has achieved so far. This value is called *pBest*. The *pBest* of the ith particle is represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$. At each iteration, the P vector of the particle with the best fitness in the neighborhood, designated l or g, and the P vector of the current particle are combined to adjust the velocity along each dimension, and that velocity is then used to compute a new position for the particle. The portion of the adjustment to the velocity influenced by the individual's previous best position (P) is considered as the *cognition* component, and the portion influenced by the best in the neighborhood is the *social* component. With the addition of the inertia factor ω, by Shi and Eberhart [59] (brought in for balancing the global and the local search), these equations are:

$$v_{id} = \omega^* v_{id} + \eta_x^* rand()^*(p_{id} - x_{id}) + \eta_2^* Rand()^*(p_{gd} - x_{id}) \qquad (1.1)$$

$$x_{id} = x_{id} + v_{id} \qquad (1.2)$$

where *rand()* and *Rand()* are two random numbers independently generated within the range [0,1] and η_1 and η_2 are two learning factors which control the influence of the social and cognitive components. In (1.1), if the sum on the right side exceeds a constant value, then the velocity on that dimension is assigned to be $\pm V_{max}$. Thus, particles' velocities are clamped to the range [$-V_{max}$, V_{max}] which serves as a constraint to control the global exploration ability of particle swarm. Thus, the likelihood of particles leaving the search space is reduced. Note that this is not to restrict the values of X_i within the range [$-V_{max}$, V_{max}]; it only limits the maximum distance that a particle will move during one iteration ([19], [20], [21]). The main PSO algorithm as described by Pomeroy [52] is given below:

/ set up particles' next location */*
for each particle p **do** {
    for $d = 1$ **to** dimensions **do** {
        p.next[d] = random()
        p.velocity[d] = random(deltaMin, deltaMax)

```
        }
        p.bestSoFar = initialFitness
    }
    /* set particles' neighbors */
    for each particle p do {
        for n = 1 to numberOfNeighbors do {
            p.neighbor[n] =getNeighbor(p,n)
        }
    }
    /* run Particle Swarm Optimizer */
    while iterations ≤ maxIterations do {
    /* Make the "next locations" current and then*/
    /* test their fitness. */
        for each particle p do {
            for d = 1 to dimensions do {
                p.current[d] = p.next[d]
            }
            fitness = test(p)
            if fitness > p.bestSoFar then do {
                p.bestSoFar = fitness
                for d = 1 to dimensions do {
                    p.best[d] = p.current[d]
                }
            }
            if fitness = targetFitness then do {
                ...
                /* e.g., write out solution and quit */
            }
        } /* end of: for each particle p */
        for each particle p do {
            n = getNeighborWithBestFitness(p)
            for d = 1 to dimensions do {
                iFactor = iWeight * random(iMin, iMax)
                sFactor = sWeight * random(sMin, sMax)
                pDelta[d] = p.best[d] - p.current[d]
                nDelta[d] = n.best[d] - p.current[d]
                delta = (iFactor * pDelta[d]) + (sFactor * nDelta[d])
                delta = p.velocity[d] + delta
                p.velocity[d] = constrict(delta)
                p.next[d] = p.current[d] + p.velocity[d]
            }
        } /* end of: for each particle p */
    } /* end of: while iterations ≤ maxIterations */
```

end */* end of main program */

```
/* Return neighbor n of particle p */
function getNeighbor(p, n) {
    ...
    return neighborParticle
}
```

```
/* Return particle in p's neighborhood */
/* with the best fitness */
function getNeighborWithBestFitness(p) {
    ...
    return neighborParticle
}
```

```
/* Limit the change in a particle's */
/* dimension value */
function constrict(delta) {
    if delta < deltaMin then
        return deltaMin
    else
        if delta > deltaMax then
            return deltaMax
        else
            return delta
}
```

The basic scheme of PSO algorithm is presented in Figure 1.4. The PSO algorithm can be seen as a set of vectors whose trajectories oscillate around a region defined by each individual previous best position and the best position of some other individuals [34]. There are different neighborhood topologies used to identify which particles from the swarm can influence the individuals. The most common ones are known as the *gbest* and *lbest:*

In the *gbest* swarm, the trajectory of each individual (particle) is influenced by the best individual found in the entire swarm. It is assumed that *gbest* swarms converge fast, as all the particles are attracted simultaneously to the best part of the search space. However, if the global optimum is not close to the best particle, it may be impossible for the swarm to explore other areas and, consequently, the swarm can be trapped in a local optima [35].

In the *lbest* swarm, each individual is influenced by a smaller number of its neighbors (which are seen as adjacent members of the swarm array). Typically, *lbest* neighborhoods comprise of two neighbors: one on the right side and one on the left side (a ring lattice). This type of swarm will converge slower but can locate the global optimum with a greater chance. *lbest* swarm is able to flow around local optima, sub-swarms being able to explore different optima [35]. A graphical representation of a
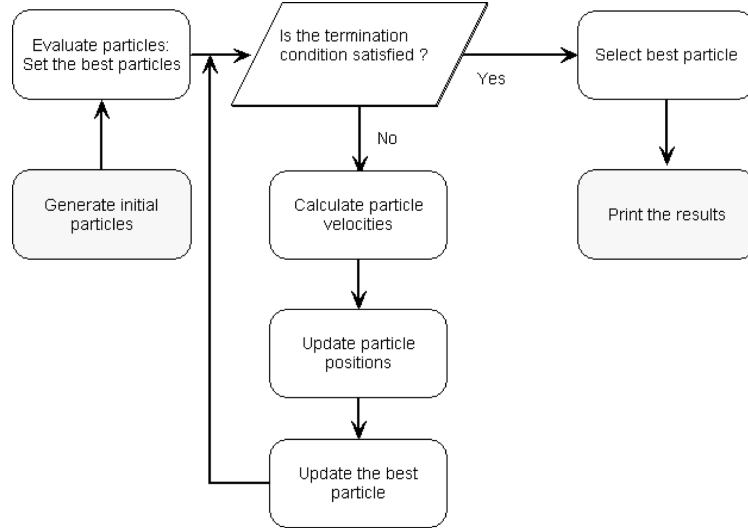
Fig. 1.4. The basic structure of PSO.

gbest swarm and a *lbest* swarm respectively is depicted in Figure 1.5 (taken from [35]). If we consider social and geographical neighborhoods as presented in Figure 1.6, then both *gbest* and *lbest* may be viewed as forms of social neighborhoods.
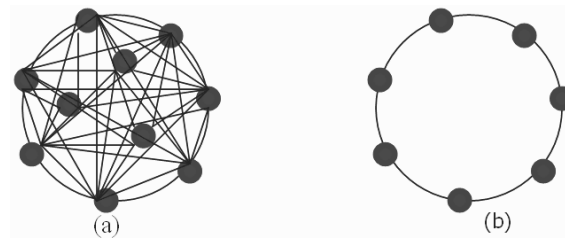
Fig. 1.5. Graphical representation of (a) *gbest* swarm (b) *lbest* swarm.

Watts [70], [71] introduced the small-world network model which allows to interpolate between regular low-dimensional lattices and random networks, by introducing a certain amount of random long-range connections into an initially regular network [14]. Starting from here, several models have been developed: icing model [5], spreading of epidemics [44], [45], evolution of random walks [27] are some of them. Watts identifies two factors influencing the information exchange between the small-world network members:

- the degree of connectivity : the behavior of each individual will be influenced by the behavior of its k neighbors.
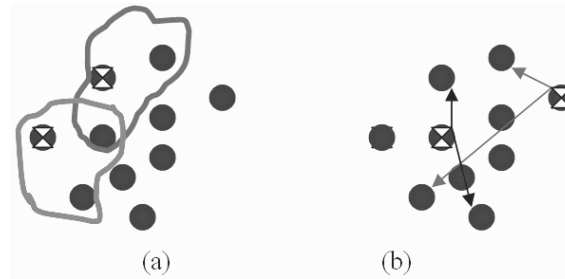
Fig. 1.6. Examples of (a) geographical neighborhood (b) social neighborhood in a swarm.

- the amount of clustering : clustering refers to the neighbors in common with some other individuals.

There are several types of swarm topologies which can be used depending upon the problem to be solved. Kennedy and Mendes [35] have tested few topologies like pyramid model, star, "small", von Neumann etc. for some function optimization problems.

1.2.2 Ant Colonies Optimization

Ant Colonies Optimization (ACO) algorithms were introduced around 1990 [15], [16], [17]. These algorithms were inspired by the behavior of ant colonies. Ants are social insects, being interested mainly in the colony survival rather than individual survival. Of interests is ants' ability to find the shortest path from their nest to food. This idea was the source of the proposed algorithms inspired from ants' behavior.

When searching for food, ants initially explore the area surrounding their nest in a random manner. While moving, ants leave a chemical pheromone trail on the ground. Ants are guided by pheromone smell. Ants tend to choose the paths marked by the strongest pheromone concentration . When an ant finds a food source, it evaluates the quantity and the quality of the food and carries some of it back to the nest. During the return trip, the quantity of pheromone that an ant leaves on the ground may depend on the quantity and quality of the food. The pheromone trails will guide other ants to the food source. The indirect communication between the ants via pheromone trails enables them to find shortest paths between their nest and food sources. As given by Dorigo et al. [18] the main steps of the ACO algorithm are given below:

1. *pheromone trail initialization*
2. *solution construction using pheromone trail*
   Each ant constructs a complete solution to the problem according to a probabilistic
3. state transition rule. The state transition rule depends mainly on the state of the pheromone [64].
4. *pheromone trail update.*

A global pheromone updating rule is applied in two phases. First, an evaporation phase where a fraction of the pheromone evaporates, and then a reinforcement phase where each ant deposits an amount of pheromone which is proportional to the fitness of its solution [64]. This process is iterated until a termination condition is reached.

1.3 Data mining

Historically the notion of finding useful patterns in data has been given a variety of names including data mining, knowledge extraction, information discovery, and data pattern processing. Data mining is the application of specific algorithms for extracting patterns from data [22]. The additional steps in the KDD process, such as data selection, data cleaning, incorporating appropriate prior knowledge, and proper interpretation of the results are essential to ensure that useful knowledge is derived form the data.
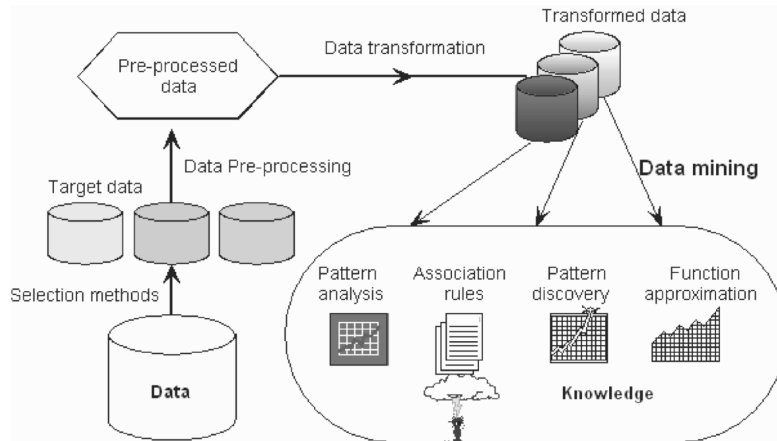
Fig. 1.7. Steps of the knowledge discovery process

1.3.1 Steps of Knowledge Discovery

Here we broadly outline some of its basic steps of the data mining process as illustrated in Figure 1.7 [22], [2].

1. *Developing and understanding the application domain, the relevant prior knowledge, and identifying the goal of the KDD process.*
2. *Creating target data set.*

3. *Data cleaning and preprocessing*: basic operations such as the removal of noise, handling missing data fields.
4. *Data reduction and projection*: finding useful features to represent the data depending the goal of the task. Using dimensionality reduction or transformation methods to reduce the effective number of variables under consideration or to find invariant representation of data.
5. *Matching the goals of the KDD process to a particular* **data mining method**: Although the boundaries between prediction and description are not sharp, the distinction is useful for understanding the overall discovery goal.

   The goals of knowledge discovery are achieved via the following data mining methods:

   - **Clustering**: identification of a finite set of categories or clusters to describe the data.
   - **Summation**: finding a compact description for subset of data, e.g. the derivation of summary for association of rules and the use of multivariate visualization techniques.
   - **Dependency modeling**: finding a model which describes significant dependencies between variables.
   - **Regression**: learning a function which maps a data item to a real-valued prediction variable and the discovery of functional relationships between variables.
   - **Classification**: learning a function that classifies a data item into one of several predefined classes.
   - **Change and Deviation Detection:** discovering the most significant changes in the data from previously measured or normative values.

1.4 Swarm Intelligence and Knowledge Discovery

Data mining and particle swarm optimization may seem that they do not have many properties in common. However, they can be used together to form a method which often leads to the result, even when other methods would be too expensive or difficult to implement. Omran [47], [48] has used particle swarm optimization methods for pattern recognition and image processing. A new clustering method based on PSO is proposed and is applied to unsupervised classification and image segmentation. The PSO-based approaches are proposed to tackle the color image quantization and spectral unmixing problems.

Visual data mining via the construction of virtual reality spaces for the representation of data and knowledge, involves the solution of optimization problems. Valdes [68] introduced a hybrid technique based on particle swarm optimization (PSO) combined with classical optimization methods. This approach is applied to very high dimensional data from microarray gene expression experiments in order to understand the structure of both raw and processed data. Experiments with data sets corresponding to Alzheimer's disease show that high quality visual representation can be obtained by combining PSO

with classical optimization methods. The behavior of some of the parameters controlling the swarm evolution was also studied.

Sousa et al. [61], [62] have proposed the use of PSO as a tool for data mining. In order to evaluate the usefulness of PSO for data mining, an empirical comparison of the performance of three variants of PSO with another evolutionary algorithm (Genetic Algorithm), in rule discovery for classification tasks is used. Such tasks are considered core tools for decision support systems in a widespread area, ranging from the industry, commerce, military and scientific fields. The data sources used here for experimental testing are commonly used and considered as a de facto standard for rule discovery algorithms reliability ranking. The results obtained in these domains seem to indicate that PSO algorithms are competitive with other evolutionary techniques, and can be successfully applied to more demanding problem domains.

Recommender systems are new types of internet-based software tools, designed to help users to find their way through today's complex on-line shops and entertainment websites. Ujjin and Bentley [66], [67] have described a new recommender system, which employs a particle swarm optimization (PSO) algorithm to learn personal preferences of users and provide tailored suggestions. Experiments are carried out to observe the performance of the system and results are compared to those obtained from the genetic algorithm (GA) recommender system and a standard, non-adaptive system based on the Pearson algorithm [7].

Another very important application of PSO is in the domain of cascading classifiers. *Cascading classifiers* have been used to solve pattern recognition problems in the last years. The main motivations behind such a strategy are the improvement of classification accuracy and the reduction of the complexity. The issue of class-related reject thresholds for cascading classifier systems is an important problem. It has been demonstrated in the literature that class-related reject thresholds provide an error-reject trade-off better than a single global threshold. Oliveira, Britto and Sabourin [46] proposed the using of the PSO for finding thresholds in order to improving the error-reject trade-off yielded by class-related reject thresholds. It has been proved to be very effective in solving real valued global optimization problems. In order to show the benefits of such an algorithm, they have applied it to optimize the thresholds of a cascading classifier system devoted to recognize handwritten digits. In a cascading classifier the inputs rejected by the first stage are handled by the next ones using costlier features or classifiers.

Settles and Rylander [56] have proposed a PSO method for neural network training. Chen and Abraham [8] investigated how the seemingly chaotic behavior of stock markets could be well represented using several soft computing techniques. Authors considered the flexible neural tree algorithm, a wavelet neural network, local linear wavelet neural network and finally a feed-forward artificial neural network. The parameters of the different learning techniques are optimized by the PSO approach. Experiment results reveal that PSO could play an important role to fine tune the parameters for optimal performance.

Breast cancer is one of the major tumor related cause of death in women. Various artificial intelligence techniques have been used to improve the diagnoses procedures and to aid the physician's efforts. Chen and Abraham [9] reported a preliminary study to detect breast cancer using a Flexible Neural Tree (FNT), Neural Network (NN), Wavelet Neural Network (WNN) and their ensemble combination. For the FNT model, a tree-structure based evolutionary algorithm and PSO are used to find an optimal FNT. For the NN and WNN, the PSO is employed to optimize the free parameters. The performance of each approach is evaluated using the breast cancer data set. Simulation results show that the obtained FNT model has a fewer number of variables with reduced number of input features and without significant reduction in the detection accuracy. The overall accuracy could be improved by using an ensemble approach by a voting method.

Chen et al. [10] proposed an evolutionary procedure to design hierarchical or multilevel fuzzy system Takagi-Sugeno Fuzzy Systems (TS-FS). The hierarchical structure is evolved using Probabilistic Incremental Program Evolution (PIPE) with specific instructions. The fine tuning of the if-then rules parameters encoded in the structure is accomplished using PSO. The proposed method interleaves both PIPE and PSO optimizations. The new method results in a smaller rule-base and good learning ability. The proposed hierarchical TS-FS is evaluated by using some forecasting problems. When compared to other hierarchical TS-FS, the proposed hybrid approach exhibits competing results with high accuracy and smaller size of the hierarchical architecture.

Skopos et al. [60] have proposed a PSO method for locating periodic orbits in a three-dimensional (3D) model of barred galaxies. Method developed an appropriate scheme that transforms the problem of finding periodic orbits into the problem of detecting global minimizers of a function, which is defined on the Poincaré surface section of the Hamiltonian system. By combining the PSO method with deflection techniques, they succeeded in tracing systematically several periodic orbits of the system.

Cluster analysis has become an important technique in exploratory data analysis, pattern recognition, machine learning, neural computing, and other engineering. The clustering aims at identifying and extracting significant groups in underlying data. The four main classes of clustering algorithms are partitioning methods, hierarchical methods, density based clustering and grid-based clustering. *Document clustering* is a fundamental operation used in unsupervised document organization, automatic topic extraction, and information retrieval. Fast and high-quality document clustering algorithms play an important role in effectively navigating, summarizing, and organizing information. Recent studies have shown that partitional clustering algorithms are more suitable for clustering large datasets due to their relatively low computational requirements [63], [73]. In the field of clustering, K-means algorithm is the most popularly used algorithm to find a partition that minimizes mean square error (MSE) measure. Although K-means is an extensively useful clustering algorithm, it suffers from several drawbacks. The objective function of the K-means is not convex

and hence it may contain local minima. Consequently, while minimizing the objective function, there is possibility of getting stuck at local minima (also at local maxima and saddle point) [55]. The performance of the K-means algorithm depends on the initial choice of the cluster centers. Besides, the Euclidean norm is sensitive to noise or outliers. Hence K-means algorithm should be affected by noise and outliers [72], [28]. In addition to the K-means algorithm, several algorithms, such as Genetic Algorithm (GA) [28], [53] and Self-Organizing Maps (SOM) [43], have been used for document clustering. Cui et al. [12] proposed a PSO based hybrid document clustering algorithm. The PSO clustering algorithm performs a globalized search in the entire solution space. In the experiments, they applied the PSO, K-means and a hybrid PSO clustering algorithm on four different text document datasets. The results illustrate that the hybrid PSO algorithm can generate more compact clustering results than the K-means algorithm.

Swarming agents in networks of physically distributed processing nodes may be used for data acquisition, data fusion, and control applications. An architecture for active surveillance systems in which simple mobile agents collectively process real time data from heterogeneous sources at or near the origin of the data is used. The system requirements are motivated with the needs of a surveillance system for the early detection of large-scale bioterrorist attacks on a civilian population, but the same architecture is applicable to a wide range of other domains. The pattern detection and classification processes executed by the proposed system emerge from the coordinated activities of agents of two populations in a shared computational environment. Detector agents draw each other's attention to significant spatiotemporal patterns in the observed data stream. Classifier agents rank the detected patterns according to their respective criterion. The resulting system-level behavior is adaptive and robust.

Ye and Chen [24] introduced an evolutionary PSO learning-based method to optimally cluster N data points into K clusters. The hybrid PSO and K-means, with a novel alternative metric algorithm is called Alternative KPSO-clustering (AKPSO)method. This is developed to automatically detect the cluster centers of geometrical structure data sets. In AKPSO algorithm, the special alternative metric is considered to improve the traditional K-means clustering algorithm to deal with various structure data sets. Simulation results compared with some well-known clustering methods demonstrate the robustness and efficiency of the novel AKPSO method.

In the literature, there are some works related to co-evolutionary Particle Swarm Optimization (Co-PSO) [40], [58], [1]. According to Shi and Krohling [58], [37] each population is run using the standard PSO algorithm, using the other population as its environment [1]. Preliminary results demonstrated that Co-PSO constitutes a promising approach to solve constrained optimization problems. The problem is the difficulty to obtain fine tuning of the solution using a uniform distribution.

1.5 Ant Colony Optimization and Data mining

Ant colony based clustering algorithms have been first introduced by Deneubourg et al. [13] by mimicking different types of naturally-occurring emergent phenomena. Ants gather items to form heaps (clustering of dead corpses or cemeteries) observed in the species of *Pheidole Pallidula* and *Lasius Niger*. The basic mechanism underlying this type of aggregation phenomenon is an attraction between dead items mediated by the ant workers: small clusters of items grow by attracting workers to deposit more items. It is this positive and auto-catalytic feedback that leads to the formation of larger and larger clusters.

The general idea for data clustering is that isolated items should be picked up and dropped at some other location where more items of that type are present. Ramos et al. [54] proposed *ACLUSTER* algorithm to follow real ant-like behaviors as much as possible. In that sense, bio-inspired spatial transition probabilities are incorporated into the system, avoiding randomly moving agents, which encourage the distributed algorithm to explore regions manifestly without interest. The strategy allows guiding ants to find clusters of objects in an adaptive way.

In order to model the behavior of ants associated with different tasks (dropping and picking up objects), the use of combinations of different response thresholds was proposed. There are two major factors that should influence any local action taken by the ant-like agent: the number of objects in its neighborhood, and their similarity. Lumer and Faieta [41] used an average similarity, mixing distances between objects with their number, incorporating it simultaneously into a response threshold function like the algorithm proposed by Deneubourg et al. [13].

Admane et al. [4], presented *AntPart*, which is an exclusive unsupervised classification technique inspired by the behavior of a particular species of ants called *Pachycondyla apicalis*. The performances of this method were compared with those of three other ones, also inspired by the social behavior of ants: *AntClass, AntTree* and *AntClust*.

Kuo et al. [4], [38] proposed ant K-means (AK) clustering method. AK algorithm modifies the K-means as locating the objects in a cluster with the probability, which is updated by the pheromone, while the rule of updating pheromone is according to total within cluster variance (TWCV).

Tsai et al. [65] proposed a novel clustering method called ant colony optimization with different favor algorithm which performed better than the fast self-organizing map (SOM) K-means approach and genetic K-means algorithm.

Weng et al. [69] proposed a time series segmentation algorithm based on the ant colony optimization algorithm to exhibit the changeability of the time series data. Authors used the Bottom-Up method, which has been reported to give good results for time series segmentation. The research result shows that time series segmentation run by the ACO algorithm not only automatically identifies the number of segments, but its segmentation cost was lower than that of the time series segmentation using the Bottom-Up method.

Shelokar et al. [57] developed an ant colony optimization metaheuristic as a rule based machine learning method, called as ant colony classifier system , and applied to three process engineering examples. The learning algorithm addresses the problem of knowledge acquisition in terms of rules from example cases by developing and maintaining the knowledge base through the use of simple mechanism, pheromone trail information matrix and use of available heuristic information. The performance of an ant colony classifier is compared with the well-known decision tree based C4.5 algorithm in terms of the predictive accuracy on test cases and the simplicity of rules discovered.

Handl et al. [10] proposed a novel ant based clustering method by incorporating adaptive, heterogeneous ants, a time-dependent transporting activity, and a method that transforms the spatial embedding produced by the algorithm into an explicit partitioning. Empirical results demonstrate the ability of ant-based clustering and sorting to automatically identify the number of clusters inherent to a data collection, and to produce high quality solutions. However, the performance of the algorithm for topographic mapping was not really very good.

Web usage mining attempts to discover useful knowledge from the secondary data obtained from the interactions of the users with the Web. Web usage mining has become very critical for effective Web site management, creating adaptive Web sites, business and support services, personalization, network traffic flow analysis and so on. Abraham and Ramos [3] proposed an ant clustering algorithm to discover Web usage patterns (data clusters) and a linear genetic programming approach to analyze the visitor trends. Empirical results clearly show that ant colony clustering performs well when compared to a self organizing map (for clustering Web usage patterns).

1.6 Conclusions

In this Chapter, we introduced some of the preliminary concepts of swarm intelligence with an emphasis on particle swarm optimization and ant colony optimization algorithms. We then described the basic data mining terminologies and also illustrated some of the past and ongoing works of swarm intelligence in data mining.

References

1. Abdelbar AM, Ragab S, Mitri S (2003) Applying Co-Evolutionary Particle Swam Optimization to the Egyptian Board Game Seega. In Proceedings of The First Asian-Pacific Workshop on Genetic Programming, (S.B. Cho, N. X. Hoai and Y. Shan editors), 9-15, Canberra, Australia
2. Abonyi J., Feil B. and Abraham A. (2005), Computational Intelligence in Data Mining', Informatica: An International Journal of Computing and Informatics, Vol. 29, No. 1, pp. 3-12

3. Abraham A, Ramos V (2003) Web Usage Mining Using Artificial Ant Colony Clustering and Genetic Programming, 2003 IEEE Congress on Evolutionary Computation (CEC2003), Australia, IEEE Press, ISBN 0780378040, 1384-1391

4. Admane L, Benatchba K, Koudil M, Siad L, Maziz S (2006) AntPart: an algorithm for the unsupervised classification problem using ants, Applied Mathematics and Computation (http://dx.doi.org/10.1016/j.amc.2005.11.130)

5. Barrat A, Weight M (2000) On the properties of small-world network models. The European Physical Journal, 13:547-560

6. Blum C (2005) Ant colony optimization: Introduction and recent trends. Physics of Life Reviews, 2, 353–373

7. Breese, J.S., Heckerman, D., Kadie, C. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, pp. 43-52, 1998

8. Chen Y, Abraham A, (2006) Hybrid Learning Methods for Stock Index Modeling, Artificial Neural Networks in Finance, Health and Manufacturing: Potential and Challenges, J. Kamruzzaman, R. K. Begg and R. A. Sarker (Eds.), Idea Group Inc. Publishers, USA

9. Chen Y, Abraham A (2005) Hybrid Neurocomputing for Detection of Breast Cancer, The Fourth IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05), Japan, Springer Verlag, Germany, pp. 884-892

10. Chen Y, Peng L, Abraham A (2006) Programming Hierarchical Takagi Sugeno Fuzzy Systems, The 2nd International Symposium on Evolving Fuzzy Systems (EFS2006), IEEE Press

11. Couzin ID, Krause J, James R, Ruxton GD, Franks NR (2002) Collective Memory and Spatial Sorting in Animal Groups, Journal of Theoretical Biology, 218, 1-11

12. Cui X, Potok TE (2005) Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm, Journal of Computer Sciences (Special Issue), ISSN 1549-3636, pp. 27-33

13. Deneubourg JL, Goss S, Franks N, Franks AS, Detrain C, Chretien L (1991) The dynamics of collective sorting: Robot-like ants and ant-like robots. Proceedings of the First International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, Cambridge, MA: MIT Press, 1, 356-365

14. Dall'Asta L, Baronchelli A, Barrat A, Loreto V (2006) Agreement dynamics on small-world networks. Europhysics Letters

15. Dorigo M, Blum C (2005) Ant colony optimization theory: A survey. Theoretical Computer Science, 344(2-3), 243-278

16. Dorigo M, Di Caro G, Gambardella LM (1999) Ant algorithms for discrete optimization. Artificial Life, 5(2), 137-72

17. Dorigo M, Gambardella LM (1997) Ant colony system: A cooperative learning approach to the traveling salesman problem. IEEE Transaction on Evolutionary Computation, 1(1), 53-66

18. Dorigo M, Bonaneau E, Theraulaz G (2000) Ant algorithms and stigmergy, Future Generation Computer Systems, 16, 851-871

19. Eberhart RC, Kennedy J (1995) A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micromachine and Human Science, Nagoya, Japan, 39-43

20. Eberhart RC, Shi Y (2001) Particle swarm optimization: developments, applications and resources. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Seoul, Korea

21. Eberhart RC, Simpson PK, Dobbins RW (1996) Computational Intelligence PC Tools, Boston, MA: Academic Press Professional

22. Fayyad U, Piatestku-Shapio G, Smyth P, Uthurusamy R (1996) Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press

23. Flake G (1999) The Computational Beauty of Nature. Cambridge, MA: MIT Press

24. Fun Y, Chen CY (2005) Alternative KPSO-Clustering Algorithm, Tamkang Journal of Science and Engineering, 8(2), 165-174

25. Handl J, Knowles J, Dorigo M (2006) Ant-based clustering and topographic mapping. Artificial Life 12(1) (in press)

26. Hu X, Shi Y, Eberhart RC (2004) Recent Advences in Particle Swarm, In Proceedings of Congress on evolutionary Computation (CEC), Portland, Oregon, 90-97

27. Jasch F, Blumen A (2001) Trapping of random walks on small-world networks. Physical Review E 64, 066104

28. Jones G, Robertson A, Santimetvirul C, Willett P (1995) Non-hierarchic document clustering using a genetic algorithm. Information Research, 1(1)

29. Kennedy J, Eberhart RC (1995) Particle Swarm Optimization. In Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, IEEE Service Center, Piscataway, NJ, Vol.IV, 1942-1948

30. Kennedy J (1997) Minds and cultures: Particle swarm implications. Socially Intelligent Agents. Papers from the 1997 AAAI Fall Symposium. Technical Report FS-97-02, Menlo Park, CA: AAAI Press, 67-72

31. Kennedy J (1998) The Behavior of Particles, In Proceedings of 7th Annual Conference on Evolutionary Programming, San Diego, USA

32. Kennedy J (1997) The Particle Swarm: Social Adaptation of Knowledge. In Proceedings of IEEE International Conference on Evolutionary Computation, Indianapolis, Indiana, IEEE Service Center, Piscataway, NJ, 303-308

33. Kennedy J (1992) Thinking is social: Experiments with the adaptive culture model. Journal of Conflict Resolution, 42, 56-76

34. Kennedy J, Eberhart R (2001) Swarm Intelligence, Morgan Kaufmann Academic Press

35. Kennedy J, Mendes R (2002) Population structure and particle swarm performance. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), 1671-1676

36. Krause J, Ruxton GD (2002) Living in Groups. Oxford: Oxford University Press

37. Krohling RA, Hoffmann F, Coelho LS (2004) Co-evolutionary Particle Swarm Optimization for Min-Max Problems using Gaussian Distribution. In Proceedings of the Congress on Evolutionary Computation 2004 (CEC'2004), Portland, USA, volume 1, 959-964

38. Kuo RJ, Wang HS, Hu TL, Chou SH (2005) Application of ant K-means on clustering analysis, Computers & Mathematics with Applications, Volume 50, Issues 10-12, 1709-1724

39. Liu Y, Passino KM (2000) Swarm Intelligence: Literature Overview, http://www.ece.osu.edu/ passino/swarms.pdf

40. Lovbjerg M, Rasmussen TK, Krink T (2001) Hybrid Particle Swarm Optimiser with Breeding and Subpopulations. Proc. of the third Genetic and Evolutionary Computation Conference (GECCO-2001), volume 1, 469-476

41. Lumer ED, Faieta B (1994) Diversity and Adaptation in Populations of Clustering Ants. Clio D, Husbands P, Meyer J and Wilson S (Eds.), Proceedings of the Third International Conference on Simulation of Adaptive Behaviour: From Animals to Animats 3, Cambridge, MA: MIT Press, 501-508

42. Major PF, Dill LM (1978) The three-dimensional structure of airborne bird flocks. Behavioral Ecology and Sociobiology, 4, 111-122

43. Merkl D (2002) Text mining with self-organizing maps. Handbook of data mining and knowledge, Oxford University Press, Inc. New York, 903-910

44. Moore C, Newman MEJ (2000) Epidemics and percolation in small-world networks. Physics. Review. E 61, 5678-5682
45. Newman MEJ, Jensen I, Ziff RM (2002) Percolation and epidemics in a two-dimensional small world, Physics Review, E 65, 021904
46. Oliveira LS, Britto AS Jr., Sabourin R (2005) Improving Cascading Classifiers with Particle Swarm Optimization, International Conference on Document Analysis and Recognition (ICDAR 2005), Seoul, South Korea, 570-574
47. Omran, M. Particle Swarm optimization methods for pattern Recognition and Image Processing, Ph.D. Thesis, University of Pretoria, 2005
48. Omran, M., Salman, A. and Engelbrecht, A. P. Image classification using particle swarm optimization. Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002), Singapore. pp. 370-374, 2002
49. Paredis J (1994) Steps towards coevolutionary classification neural networks, Artificial Life IV, MIT Press, 359-365
50. Partridge BL, Pitcher TJ (1980) The sensory basis of fish schools: relative role of lateral line and vision. Journal of Comparative Physiology, 135, 315-325
51. Partridge BL (1982) The structure and function of fish schools. Science American, 245, 90-99
52. Pomeroy P (2003) An Introduction to Particle Swarm Optimization, http://www.adaptiveview.com/articles/ipsop1.html
53. Raghavan VV, Birchand K (1979) A clustering strategy based on a formalism of the reproductive process in a natural system. Proceedings of the Second International Conference on Information Storage and Retrieval, 10-22
54. Ramos V, Muge, F, Pina, P (2002) Self-organized data and image retrieval as a consequence of inter-dynamic synergistic relationships in artificial ant colonies. Soft Computing Systems - Design, Management and Applications, Proceedings of the 2nd International Conference on Hybrid Intelligent Systems, IOS Press, 500-509
55. Selim SZ, Ismail MA (1984) K-means Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality, IEEE Transaction on Pattern Analysis and Machine Intelligence, 6, 81-87
56. Settles M, Rylander B (2002) Neural network learning using particle swarm optimizers. Advances in Information Science and Soft Computing, 224-226
57. Shelokar PS, Jayaraman VK, Kulkarni BD (2004) An ant colony classifier system: application to some process engineering problems, Computers & Chemical Engineering, 28(9), 1577-1584
58. Shi Y, Krohling RA (2002) Co-evolutionary particle swarm optimization to solving min-max problems. In Proceedings of the IEEE Conference on Evolutionary Computation, Hawai, 1682-1687
59. Shi Y, Eberhart RC (1998) A modified particle swarm optimizer. In Proceedings of the IEEE Congress on Evolutionary Computation (CEC), Piscataway, NJ. 69-73
60. Skopos C, Parsopoulus KE, Patsis PA, Vrahatis MN (2005) Particle swarm optimization: an efficient method for tracing periodic orbits in three-dimensional galactic potential, Mon. Not. R. Astron. Soc. 359, 251-260
61. Sousa T, Neves A, Silva A (2003) Swarm Optimisation as a New Tool for Data Mining, International Parallel and Distributed Processing Symposium (IPDPS'03), 144b
62. Sousa T, Silva A, Neves A (2004) Particle Swarm based Data Mining Algorithms for classification tasks, Parallel Computing, Volume 30, Issues 5-6, 767-783
63. Steinbach M, Karypis G, Kumar V, (2000) A Comparison of Document Clustering Techniques. TextMining Workshop, KDD

64. Toksari MD (2006) Ant colony optimization for finding the global minimum. Applied Mathematics and Computation, (in press)
65. Tsai CF, Tsai CW, Wu HC, Yang T (2004) ACODF: a novel data clustering approach for data mining in large databases, Journal of Systems and Software, Volume 73, Issue 1, 133-145
66. Ujjin S, Bentley PJ (2002) Learning User Preferences Using Evolution. In Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning, Singapore
67. Ujjin S, Bentley PJ (2003) Particle swarm optimization recommender system. Proceedings of the IEEE Swarm Intelligence Symposium 2003 (SIS 2003), Indianapolis, Indiana, USA, 124-131
68. Valdes J (2004) Building Virtual Reality Spaces for Visual Data Mining with Hybrid Evolutionary-Classical Optimization: Application to Microarray Gene Expression Data. Proceedings of the IASTED International Joint Conference on Artificial Intelligence and Soft Computing (ASC'2004), 713-720
69. Weng SS, Liu YH (2006) Mining time series data for segmentation by using Ant Colony Optimization, European Journal of Operational Research, (http://dx.doi.org/10.1016/j.ejor.2005.09.001)
70. Watts DJ (1999) Small Worlds: The Dynamics of Networkds Between Order and Randomness. Princeton University Press
71. Watts DJ, Strogatz SH (1998) Collective dynamics of small-world networks. Nature, 393, 440-442
72. Wu KL, Yang MS (2002) Alternative C-means Clustering Algorithms. Pattern Recognition, 35, 2267-2278
73. Zhao Y, Karypis G (2004) Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering, Machine Learning, 55(3), 311-331