CHAPTER 4

EXPECTATION MAXIMIZATION WITH PARTICLE SWARM OPTIMIZATION BASED WEIGHTED CLUSTERING (EMPWC) FOR OUTLIER DETECTION IN LARGE-SCALE DATA

4.1. INTRODUCTION

Outlier detection can generally be seen in the form of a pre-processing step for the finding those objects which do not confirm to well-determined ideas of anticipated behavior in a data set. It is hugely significant in the field of data mining for exploring about new or infrequent occurrences, abnormalities, malicious behavior, unexpected conditions, etc. But the investigation of outlier detection for use in categorical data sets is particularly a challenge due to the hardship encountered while defining a purposeful similarity measure.

This problem is solved by using the Ascent-based Monte Carlo Expectation–Maximization (AMCEM) in chapter 3.

Here the outlier detection is now formulated in order to be expressed as a problem of optimization. For an arbitrary o, the number of likely candidate sets for the objective function is computed as

\[ C_n^O = \frac{n!}{o!(n-o)!} \]

that, in turn, is a very big value. Also, one might have to decide about the optimal value of \( o \), i.e., how much number of outliers in the original dataset becomes difficult task. A probable theoretical approach to this issue is to look out for a range of values of \( o \) and determine an optimal value of \( o \) through the optimization of a particular variational characteristic of \( J_X(X) \). For this purpose the focus is on developing Particle Swarm Optimization (PSO) to solve the optimization problem.

Proposed research comprises of three important steps: (i) define a function for outlier factor (ii) optimization value of outlier and (iii) clustering techniques for outlier detection. In the initial step of the work a novel concept of entropy which considers both Shannon and Jensen-Shannon Divergence (JSD) is defined. In the second step, PSO is proposed for searching the outliers. Here the PSO contains n
number of data samples N which are travelling around a D-dimensional search space for the optimization of a particular variational property. On the basis of this PSO, a function formulated for the outlier factor of an object is defined that is just decided by the object itself and could be efficiently updated. Finally EMPWC outlier detection technique is proposed that needs none of the user-specified parameters for determining if an object is to be considered an outlier or not. Further, this EMPWC based outlier detection techniques correlate a weight from entropy function with every dataset sample observed. Here the weighted-data Gaussian mixture and EM algorithms are introduced. The first one takes a weight for every categorical data attribute into consideration. Then the second one treats every weight and detects the outliers.

4.2. OVERVIEW OF SWARM INTELLIGENCE (SI)

In the recent times, a family of nature motivated algorithms, referred to as Swarm Intelligence (SI), has drawn the attention of many research people from the domain of pattern recognition and clustering. Clustering methodologies depending on the SI tools has found to be performing better than numerous conventional techniques of dividing a sophisticated and practical dataset.

Swarm Intelligence (SI) (Abraham et al., 2008) is a considerably innovative interdisciplinary area of research that has reached the heights of great popularity recently. Algorithms that belong to this domain, are influenced by the aggregated intelligence evolving from the behaviour seen in a set of social insects (such as bees, termites and wasps). When in a community, these insects can collaboratively (in group) carry out several complicated tasks that are required for their survival even when they are restricted by less individual potential. Challenges such as the discovery and storing of the foods, selection and picking up of materials for future utilization need elaborate planning, and are resolved by insect colonies in the absence of any type of supervisor or controller.

SI systems are usually derived from a population consisting of simple agents, entities able to perform/executes specific operations that interact locally with one
another and also their environment. Even though, in general, no central control structure that dictates on the behaviour of the individual agents exists, the local interactions between such agents frequently result in the development of global behavior. Several biological creatures like fish schools and bird rocks apparently exhibit a structural order, and the behavior of the organisms are so much coordinated that even if they transform in shape and direction, they are observed to move like a single logical entity.

The advantages of SI clustering in comparison with conventional clustering shows up in conditions where the present clustering is being used on a regular basis by a user or system. In the cases of such a setting as this, evolutionary clustering is helpful for the reasons below:

(1) Consistency: A user will notice the clustering of each day with familiarity, and therefore will not need to learn an entirely new means of segmentation of data. In a similar manner, any insights obtained from a study of earlier clusters have more possibility to be applied to future clusters.

(2) Noise removal: Yielding a high quality and historically persistent clustering yields higher reliability against noise by considering the earlier data points into account.

(3) Smoothing: In case the real clusters transition over time, evolutionary clustering will typically show the user, a smooth view about the shift.

(4) Cluster correspondence: It is usually possible to have the clusters of today placed in contact with the clusters of yesterday. Therefore, even in case the clustering has moved, still the user will be placed within the historical context.

Since traditional clustering method problem can be defined to be a problem of optimization, SI approaches might be applicable here. The concept is to make use of SI functions and a population consisting of clustering structures to meet into a globally optimal clustering. Each one of the anonymized datapoint is considered as the
swarm. A fitness function evaluation on the swarms, decides a swarm’s chances of survival into the next generation.

SI systems are typically made up of a population consisting of simple agents that interact locally with each other and their environment. Generally, no central control structure that governs the behaviour of the individual agents exists, and the local interactions observed between these agents frequently result in the evolution of a global behaviour. Few examples of systems such as this can be seen in nature, inclusive of ant colonies, bird flocking, bat, flashing behaviour, bee swarming, animal herding, bacteria moulding and fish schooling.

Issues such as discovery and storage of foods, selection and pick up of materials for future use need an elaborate planning, and are resolved by insect colonies with no supervisor or controller. An instance of some notably successful research expeditions in swarm intelligence includes Ant Colony Optimization (ACO) Dorigo and Blum (2005); Dorigo et al., (1999); Dorigo and Gambardella (1997) that highlights on problems involving discrete optimization, and has been used with success to several NP hard discrete optimization problems inclusive of the traveling salesman, the quadratic assignment, scheduling, vehicle routing, etc., in addition to routing done in telecommunication networks. Particle Swarm Optimization (PSO) Kennedy (2011) is one more widely-known SI algorithm used for global optimization over contiguous search spaces. Among this behavior the following PSO Kennedy (1997) is selected for implementation of optimizing outliers in the clustering for categorical data.

4.2.1. Particle Swarm Optimization (PSO)

PSO is a common population-based search algorithm and is initiated with a population full of random solutions, referred to as particles Kennedy (1998). In contrary to the other path-breaking computation methodologies, every particle in PSO is also related to velocity. Particles are observed to fly through the search space with velocities that are adjusted dynamically in accordance to their historical behaviours. Hence, the particles tend to fly towards the best search area during the course of search procedure. The PSO was initially developed in order to simulate the birds
looking out for food, that, in turn, is defined to be a ‘cornfield vector’ Kennedy (1997).

PSO uses the scenario for learning purposes and then for solving the optimization problems. In PSO, every single solution is similar to a ‘bird’ in the search space, referred to as ‘particle’. Each particle has fitness values that are assessed by the fitness function for optimization, and possess velocities that guide the flight of the particles. (The particles fly in the problem space by following the path of the particles having the best solutions until now). PSO is initiated with a set of random particles (solutions) and afterwards looks out for optima by updating every generation.

The fundamental scheme of PSO algorithm is presented in Figure 4.1. The PSO algorithm could be seen in the form of a group of vectors, the trajectories of which oscillate around a locationspecified by every individual’s previous best position and the best position of few other individuals Kennedy et al., (2001). There are various neighborhood topologies that are employed for identifying the particles from the swarm that can have an effect over the individuals. The most usual ones are referred to as the gbest and lbest. In the case of gbest swarm, the best individual observed in the whole swarm influences the trajectory of every individual (particle). It is an assumption that gbest swarms converge rapidly, since all the particles are attracted at the same time to the best region of the search space.

But, in case the global optimum is not near to the best particle, then it may not be possible for the swarm to investigate other regions and, as a result, the swarm can slip into local optima Kennedy and Mendes (2002). In the lbest swarm, every individual is impacted by a few number of its neighbors (that are found to beneighborhood members of the swarm group). Generally, lbest neighborhoods consists of two neighbors: one that is located on the right side and another on the left side (a ring lattice). This kind of swarm will converge slowly but its chances for finding the global optimum are more. lbest swarm is capable of flowing around the local optima, and subswarms are capable of exploring diverse optima Kennedy and
Mendes (2002). A gbest swarm and an lbest swarm are graphically represented in Figure 4.2 (obtained from Kennedy and Mendes (2002)).

Figure 4.1. The basic structure of PSO
The PSO clustering algorithm carries out a global search in the whole solution space.

4.3. GAUSSIAN MIXTURE MODEL (GMM) WITH EXPECTATION MAXIMIZATION

The conventional EM algorithm arbitrarily selects the samples to be at the center of every class that easily has an influence over the result of clustering. Otherwise said, the drawback of conventional EM algorithm is that it is highly dependent on the choice of the initial center of every class. Moreover, the marginal values have a greatly possibility of having an effect over the complete algorithm, thus reducing the accuracy.

The Gaussian Mixture Model (GMM) is combined with the EM algorithm Elkan (1997) in order to alter the data model into a low-dimensional one to reduce complexity. GMM is generally a Gaussian probability density function that does the accurate quantification of things, and things will be divided into a number of models on the basis of the generation of Gaussian probability density function. Modeling process requires few parameters of the Gaussian mixture model, like variance, mean, weights and few other initiation parameters, and then exploits these parameters received by modeling the data required.

By updating the expectation and coefficient, a new Gaussian distribution density model is obtained for the next iteration, with the updated expectation and coefficient values as the input in the next Expectation Step. The Expectation Step and
Maximization Step are repeated till the entire program attains convergence; which indicates that, when the variation in the expectation and coefficient values is adequately small.

### 4.4. PROPOSED EXPECTATION MAXIMIZATION WITH PARTICLE SWARM OPTIMIZATION BASED WEIGHTED CLUSTERING (EMPWC) FOR OUTLIER DETECTION

In this work, a formal optimization-based model of categorical outlier detection is proposed, which uses an innovative concept concerned with Shannon entropy that acquires the distribution of a dataset. The newly introduced work consists of three major steps: (i) specify a function for outlier factor (ii) optimization value of outlier and (iii) clustering methods for outlier detection. In the first step of the work define a novel concept of entropy, which takes both Shannon and Jensen-Shannon Divergence (JSD) into consideration. Second step PSO is introduced to search outliers. On the basis of this PSO, a function for the outlier factor of an object is presented that, in turn, is only decided by the object itself and can be updated with efficiency. At finally EMPWC outlier detection method is proposed which requires no user-specified parameters to decide if an object is an outlier or not. In addition this EMPWC based outlier detection methods that associate a weight from entropy function with each observed dataset samples. Here introduce the weighted-data Gaussian mixture and EM algorithms. The first one considers a weight for each categorical data attributes. The second one treats each weight and detects outliers.

#### 4.4.1. Measurement for Outlier Detection

Consider an data be the $X$ containing number of the data objects as $n(x_1, ..., x_n)$ each $x_i$ for $1 < i < n$ being a vector of categorical attributes $[y_{1j}, y_{2j}, ..., y_{mj}]^T$, where $m$ refers to the number of categorical and discrete data attributes. $y_j$ represents the value of the attribute that belongs to either categorical and discrete value represented by $(y_{1j}, y_{2j}, ..., y_{nj})(1 < j < m)$ and $n_j$ refers to the number of unique values present in attribute $y_j$. For the purpose of measuring the attribute
value importance, the Shannon, Jensen-Shannon Divergence (JSD) is applied and the holoentropy of the attribute is also measured to categorical attributes.

### 4.4.2. Shannon Entropy and Jensen-Shannon Divergence (JSD)

Shannon entropy is one among the most essential metrics in information theory. Entropy does the measurement of the uncertainty which is associated with a stochastic variable, i.e. the anticipated value of the data present in the message (in conventional informatics it is measured in terms of bits).

\[
H(X) = \sum_{i=1}^{n} p(x_i) \log_b \frac{1}{p(x_i)} \tag{4.1}
\]


\[
JS(y_i|y_j) = \frac{1}{2} \sum_{i} p(y_i) \ln \frac{P(y_i)}{\frac{1}{2} (P(y_i) + P(y_j))} + \frac{1}{2} \sum_{j} p(y_j) \ln \frac{P(y_j)}{\frac{1}{2} (P(y_i) + P(y_j))} \tag{4.2}
\]

\[
= \frac{1}{2} D(y_i||M) + \frac{1}{2} D(y_j||M) = S(M) - \frac{1}{2} S(y_i) - \frac{1}{2} S(y_j) \tag{4.3}
\]

The equation (4.4) provides the probability computation formula of every firefly for a set of data given.

\[
p(y_i) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{V_n} \varphi \left( \frac{y_i - y_m}{h_n} \right) \tag{4.4}
\]

where \( \varphi(x)\) stands for the window function and \( n \) refers to the total number of data objects, \( V_n \) and \( h_n \) are the respective volume and edge length of a hypercube. When the JSD is computed, then the weights are computed from the data directly and is influenced by the increase in the effectiveness in practical applications compared to theoretical requirement.
The weighted holoentropy of random vector $W_X(Y)$ is defined to be the summation of the weighted entropy on every attribute of the random vector $Y$.

$$W_X(y) = \sum_{i=1}^{m} w_x(y_i) H_X(y_i)$$  \hspace{1cm} (4.6)

Provided a data set $X$ with $n$ objects and the number $o$, a subset $\text{Out}(o)$ is defined to be the set of outliers in case it reduces $J_X(Y; o)$, defined to be the weighted holoentropy of $X$ having $o$ objects eliminated

$$J_X(Y, O) = W_{X\setminus \text{set}(O)}(Y)$$  \hspace{1cm} (4.7)

where set(O) refers to any subset of $o$ objects from $X$. To be otherwise said

$$\text{Out}(O) = \text{argmin} J_X(Y, O)$$  \hspace{1cm} (4.8)

Therefore, the formulation of the outlier detection is now expressed to be a problem of optimization. For a provided $o$, the number of probable candidate sets for the objective function is $C_n^o = \frac{n!}{o!(n-o)!}$, that is extremely huge. In addition, one may need to decide the optimal value of $O$, i.e., the number of outliers a data set actually.

A probable theoretical scheme used to this issue is searching for a range of values of $O$ and then deciding over an optimal value of $O$ through the optimization of a particular variational property of $J_X(Y, O)$. Assume this to be a direction proposed in this research work. At present, focus will be on the development of practical solutions for the optimization issue.

**4.4.3. Particle Swarm Optimization (PSO)**

PSO is generally a type of optimization tool that is population independent, presented initially in the form of an optimization approach to be used for
real-number spaces. In the scheme of PSO, each particle has analogy to an individual “fish” that is present in a group of fish. In this, for choosing the most optimizing a particular variational property of $J_X(Y,O)$ analysis and optimization for range of values for $O$. PSO contains $n$ number of data samples $N$ which move around a D-dimensional search space for the optimization of a particular variational property of $J_X(Y,O)$.

The process of PSO is begin with a population that consists of a number of the data objects as $n(x_1, ..., x_n)$ with every $x_i$ with $r_1 \ldots r_i$ refer to the attribute numbers selected from 1 to $m$ for each data sample and the optimization appropriately next searches for the best range of values for $O$ through continuously updating the generations. Each data object (particle) uses its individual data object which having from at the same cluster among data points.

The knowledge which is gained by the swarm completely to discover the optimization of a particular variational property of $J_X(Y,O)$ in a cluster. The location of the $i^{th}$ data samples of cluster particle can be referred to by $l = (l_1, \ldots, l_j)$. The velocity with respect to the $i^{th}$ cluster of data points could be represented as $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$. The velocities corresponding to the data points in the cluster are restricted within $[V_{min}, V_{max}]^D$, correspondingly.

The best earlier visited location of the $i^{th}$ data points denoted its individual best outlier detection results $l_{best} = (l_{b1}, l_{b2}, \ldots, l_{bD})$, a value known as $l_{best}$. The best value of the whole individual $l_{best}$ values is represented the global best position $g_{best} = (g_{b1}, g_{b2}, \ldots, g_{bD})$ and referred to as $g_{best}$. During every generation, the position and the velocity of each $i^{th}$ data points in the cluster gets reviewed by $l_{best}$ and $g_{best}$ present in the swarm. It occurs in the space which data point’s discrete problem, with the intent of resolving this issue, PSO which is used for discrete binary variables.

In binary space, a particle which is a data point in the cluster probably will transition to the nearly corners of a hypercube through the flipping of multiple numbers of bits; and consequently, the particle velocity on the whole may be defined
with the number of bits modified according to the number of processes. In PSO, each
data point for outlier’s elimination get revised as per the equations as below:

\[
v_{id}^{new} = w \times v_{id}^{new} + c_1 \times r_1 \times (lbest_{id} - O_{id}^{old}) + c_2 \times r_2 \times (gbest_{id} - O_{id}^{old}) \quad (4.9)
\]

If \( v_{id}^{new} \notin (V_{min}, V_{max}) \) then \( v_{id}^{new} = \max(\min(V_{max}), v_{id}^{new}), V_{min} \)

\[
S(v_{id}^{new}) = \frac{1}{1 + e^{-v_{id}^{new}}} \quad (4.10)
\]

If \( r_3 < S(v_{id}^{new}) \) then \( O_{id}^{new} = 1 \) else \( O_{id}^{new} = 0 \) \( (4.11) \)

Here, \( w \) refers to the inertia weight for the optimization of a particular
variational property of \( J_X(Y, O) \) its current dataset samples, \( r_1, r_2 \) and \( r_3 \) refer to the
random numbers within \((0, 1)\), and \( c_1 \) and \( c_2 \) indicate the acceleration constants.
Velocities \( v_{id}^{new} \) and \( v_{id}^{old} \) refer to the new and old velocities for outliers. \( v_{id}^{old} \) indicates
the current particle position, and \( v_{id}^{new} \) refers to the new, updated outlier detection position.

In Equation (4.11), outlier detection position velocities of every dataset
sample are attempted to be at a maximum velocity \( V_{max} \). Once the total of
accelerations renders the velocity of that specific dimension to move above \( V_{max} \),
consequently the velocity of that particular dimension is restricted to \( V_{max} \).
\( V_{max} \) and \( V_{min} \) indicate the limitations. If \( S(v_{id}^{new}) \) is greater than \( r_3 \), consequently its position value of the current data point is represented by \( \{1\} \) somewhere else \( \{0\} \).

When the outlier values get optimized the clustering process becomes easy
to perform. At last the clustering is carried out employing Expectation Maximization (EM) for PSO based Weighted Clustering (EMPWC) algorithm. Earlier experiment shows that the performance of precise and approximate outlier factor has huge similarity. To prevent the large time complexity of accurate computation of factor, the approximate factor \( J_X(Y, O) \) is used for representing the approximate one in this technical work.
4.4.4. Expectation Maximization (EM) for PSO based Weighted Clustering (EMPWC)

Discovering important groups in a collection of data points is a major issue in several areas. As a result, clustering has gained much attention, and several techniques, algorithms and software packages are available in the recent times. Among these methods, parametric finite mixture models have a considerable role to play, because of their interesting mathematical properties in addition to the presence of maximum likelihood estimators that is dependent on Expectation-Maximization (EM) algorithms.

With the finite Gaussian mixture (GMM) Reynolds (2015) being the model that is desired, it is greatly sensitive to the existence of outliers. Alternate robust models have been introduced in the statistical literature, like the mixtures of skew t-distributions Lin et al., (2007) and their various variants, e.g. Forbes and Wraith (2014); Lee and McLachlan (2014).

Hence in this technical work an Expectation Maximization (EM) is proposed for PSO based Weighted Clustering (EMPWC) where the variable $W_X(y)$ is deployed in the form of a weight in order to account for the robustness of the dataset samples $X_i$ observed and this independently over its cluster assigned. The distribution of $W_X(y)$ is not anymore a gamma mixture but it has to rely on $i$ to permit a very data point to be significantly treated in a different manner.

In this technical work, the weighted data is introduced with Gaussian Mixture Model in the form of two cases, i) the weights $W_X(y)$ are decided by employing factors such as holoentropy and JSD, they are predetermined, and (ii) the weights are modeled to serve as variables and therefore they are updated iteratively when the sample varies. Thereafter, on the basis of these weights to be optimal value of $O$ by the optimization of a specific variational property of $J_X(Y, O)$ making use of PSO, it is modelled to be random variables. These variables are Modelled with gamma
distributions and a closed-form EM algorithm is derived that will be indicated as the EMPWC. Afterwards, M-step and E-step are continuously updated.

In this technical work, it is also proposed to apply the weighted-data robust clustering technique for the data clustering and outlier detection problem. This section introduces the formal definition of the proposed EMPWC algorithm. Consider \( x \in \mathbb{R}^d \) to be a random data sample vector that follows a multivariate Gaussian distribution having mean \( \mu \in \mathbb{R}^d \) and covariance \( \Sigma \in \mathbb{R}^d \) namely \( p(x|\theta) = \mathcal{N}(x; \mu, \Sigma) \) with the notation \( \theta = \{ \mu, \Sigma \} \). Suppose \( W_X(y) > 0 \) refer to a weight that indicates the relevance of the attribute value measurement results from the dataset samples \( x \). In an intuitive manner, the greater the weight \( w \), the more strong will be the effect of \( x \) on cluster.

With respect to the likelihood function, this is equal to increment \( p(x; \theta) \) to the power \( W_X(y) \), i.e. \( \mathcal{N}(x; \mu, \Sigma)^{W_X(y)} \). It is pretty direct to observe that \( \mathcal{N}(x; \mu, \Sigma)^{W_X(y)} \alpha \mathcal{N}(x; \mu, \Sigma/W_X(y)) \). Hence, \( W_X(y) \) has a considerable role to play in increasing the clustering results and varies for every dataset samples \( x \). Consequently, write:

\[
\hat{p}(x, \theta, W_X(y)) = \mathcal{N}(x; \mu, \frac{1}{W_X(y)} \Sigma) \tag{4.12}
\]

from which a mixture model is derived with \( K \) components:

\[
\hat{p}(x, \theta, W_X(y)) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x; \mu_k, \frac{1}{W_X(y)} \Sigma_k) \tag{4.13}
\]

where \( \Theta = \{ \pi_1, ..., \pi_K, \theta_1, ..., \theta_K \} \) refer to the mixture parameters \( \pi_1, ..., \pi_K \) indicate the mixture coefficients that satisfy \( \pi_k \geq 0 \) and \( \sum_{k=1}^{K} \pi_k = 1 \), \( \theta_k = \{ \mu_k, \Sigma_k \} \) refer to the parameters of the \( k \)th component and \( K \) stands for the number of components. The model in (4.13) will be referred to as the weighted function obtained from equation (4.11). Consider \( X = \{ x_1, ..., x_n \} \) to be the observed data and \( W_X(y) = \{ W_1(y), ..., W_n(y) \} \) refer to the weights that are associated with \( X \). Consider that \( x_i \) to observed-data log-likelihood is:
\begin{align}
\ln \hat{p}(x, \theta, W_X(y)) &= \sum_{i=1}^{n} \ln \left( \sum_{k=1}^{K} \pi_k \mathcal{N}(x_i; \mu_k, \frac{1}{W_X(y)} \Sigma_k) \right) \tag{4.14}
\end{align}

It is common knowledge that direct maximization of the log-likelihood function is hard with regard to mixtures and that the anticipated complete-data log-likelihood has to be taken in its place. Therefore, a set consists of \( n \) hidden (assignment) variables \( Z = \{z_1, \ldots, z_n\} \) which are related to the observed variables \( X \) and in such a manner that \( z_i = k, k \in \{1, \ldots, K\} \) if and only if \( x_i \) gets created by the \( k \)-th component of the mixture. Consider a fixed (given) number of mixture components \( K \), and thereafter the model is extended to an unknown \( K \), thus making an estimate of the number of components from the data. Thus the overall expected data log-likelihood is:

\begin{align}
Q_c(\Theta, \Theta^{(r)}) &= E_P(Z|X; W_X(y), \Theta^{(r)}) \left[ \ln P(X, Z, W_X(y), \Theta) \right] \tag{4.15}
\end{align}

where \( E_P[\cdot] \) represents the expectation with regard to the distribution \( P \). Distribution \( P \). The \((r + 1)\)-th EM iteration comprises of two steps which are, the assessment of the posterior distribution with the current model parameters \( \Theta^{(r)} \), the weights \( W \), and the maximization of (4.15) with regard to \( \Theta \) (M-step):

\begin{align}
\Theta^{(r+1)} &= \arg \max_{\Theta} Q_c(\Theta, \Theta^{(r)}) \tag{4.16}
\end{align}

E-Step

The posteriors \( \eta_{ik}^{(r+1)} = p(z_i = k|x_i; W_X(y), \Theta^{(r)}) \) are updated with

\begin{align}
\eta_{ik}^{(r+1)} &= \frac{\pi_k^{(r)} \hat{p}(x, \theta^{(r)}, W_X(y))}{\hat{p}(x, \Theta^{(r)}, W_X(y))} \tag{4.17}
\end{align}

M-Step

\begin{align}
Q_c(\Theta, \Theta^{(r)}) &= \sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(r+1)} \ln \pi_k \mathcal{N}(x; \mu_k, \frac{1}{W_X(y)} \Sigma_k) \tag{4.18}
\end{align}
$$\sum_{i=1}^{n} \sum_{k=1}^{K} \eta_{ik}^{(r+1)} (\ln \pi_k - \ln |\Sigma_k|^2 - \frac{W_X(y)}{2} (x_i - \mu_k)^T \Sigma_k^{-1} (x_i - \mu_k))$$

(4.19)

$$\pi_k^{(r+1)} = \frac{1}{n} \sum_{i=1}^{n} \eta_{ik}^{(r+1)}$$

(4.20)

$$\mu_{ik}^{(r+1)} = \frac{\sum_{i=1}^{n} W_X(y) \eta_{ik}^{(r+1)} x_i}{\sum_{i=1}^{n} W_X(y) \eta_{ik}^{(r+1)}}$$

(4.21)

$$\Sigma_{ik}^{(r+1)} = \frac{\sum_{i=1}^{n} W_X(y) \eta_{ik}^{(r+1)} (x_i - \mu_{ik}^{(r+1)}) (x_i - \mu_{ik}^{(r+1)})^T}{\sum_{i=1}^{n} \eta_{ik}^{(r+1)}}$$

(4.22)

This process gets repeated till the lower bound becomes positive. The upper bound on outliers (UO), the anomaly candidate set (AS), and the normal object set (NS). This way the data objects having positive lower bound is observed as $$AS = \{x_i | D_y | y > 0\},$$ the data objects having then on positive set is observed as,

$$UO = N(AS) = \sum_{i=1}^{n} (\bar{D}_y | y_j > 0) \ arg \ max J_X(Y, 0)$$

(4.23)

Propose an efficient EMPWC technique for outlier detection.

4.5. PERFORMANCE EVALUATION

This section carries out the efficiency and effectiveness tests for analyzing the performance of the novel EMPWC technique. In order to test effectiveness, the result is compared with the available techniques such as Information-Theory-Based Step-by-Step (ITB-SS) and Information-Theory-Based Single-Pass (ITB-SP) for artificial data sets. For the test of efficiency, evaluations on are conducted over artificial data sets to indicate how execution time sees an increase with the number of objects, attributes and the outliers. A huge number of public actual data sets, many of them obtained from UCI, are utilized in these experiments, indicating an extensive range of fields in science and humanities. Area Under the Curve (AUC) and significance test are applied for measuring results are discussed in Table 4.1.
Table 4.1. AUC Results of Tested Algorithms on the Real dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#n</th>
<th>#m</th>
<th>#o</th>
<th>#UO</th>
<th>FIB</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>EMPWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Breast-c</td>
<td>495</td>
<td>11</td>
<td>45</td>
<td>125</td>
<td>0.90</td>
<td>0.991</td>
<td>0.993</td>
<td>0.996</td>
<td>0.997</td>
</tr>
<tr>
<td>Credit-a</td>
<td>413</td>
<td>17</td>
<td>30</td>
<td>171</td>
<td>0.92</td>
<td>0.985</td>
<td>0.992</td>
<td>0.995</td>
<td>0.996</td>
</tr>
<tr>
<td>Diabetes</td>
<td>768</td>
<td>9</td>
<td>268</td>
<td>340</td>
<td>0.88</td>
<td>0.75</td>
<td>0.912</td>
<td>0.945</td>
<td>0.945</td>
</tr>
<tr>
<td>Ecoli</td>
<td>336</td>
<td>8</td>
<td>9</td>
<td>144</td>
<td>0.92</td>
<td>0.96</td>
<td>0.99</td>
<td>0.996</td>
<td>0.998</td>
</tr>
</tbody>
</table>

4.5.1. Efficiency of Real Data Sets

For measuring the results of time consumption with increase in the number of objects, attributes and outliers.

![Efficiency analysis of real data sets with data objects vs. methods](image)

Figure 4.3. Efficiency analysis of real data sets with data objects vs. methods

The time consumption with rising numbers of objects, attributes and outliers is measured. Like the Figure 4.3 shows, the execution times of EMPWC, AMCEM, ITB-SP, and ITB-SS are nearly linear functions of the number of objects. From the figure 4.3 it concludes that the proposed EMPWC has lesser execution time results of 7 seconds for 200 data objects which is 1 second, 4 seconds, and 5 seconds lesser when compared to AMCEM, ITB-SS, and ITB-SP methods respectively. The
results of all the methods in terms of data objects are discussed and tabulated in table 4.2.

**Table 4.2. Time comparison results for data objects vs. methods**

<table>
<thead>
<tr>
<th>No.of dataset points</th>
<th>Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ITB-SP</td>
</tr>
<tr>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>100</td>
<td>23</td>
</tr>
<tr>
<td>150</td>
<td>18</td>
</tr>
<tr>
<td>200</td>
<td>12</td>
</tr>
</tbody>
</table>

**Figure 4.4. Efficiency analysis of real data sets with data attributes vs. methods**

Figure 4.4 illustrates that the execution times of the EMPWC and other existing methods. This increases quickly with the number of attributes increasing in a quadratic way. In comparison with the increase in time of ITB-SS, ITB-SP, AMCEM the increase in time for the other techniques are shown in Figure 4.4. From the figure 4.4 it concludes that the proposed EMPWC has lesser execution time results of 5 seconds for 30 data attributes which is 1 second, 4 seconds, and 7 seconds lesser when
compared to, AMCEM, ITB-SS and ITB-SP methods respectively. The results of all the methods in terms of data attributes are discussed and tabulated in Table 4.3.

**Table 4.3. Time comparison results for data attributes vs. methods**

<table>
<thead>
<tr>
<th>No. of attributes</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>EMPWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>20</td>
<td>18</td>
<td>12</td>
<td>11</td>
</tr>
<tr>
<td>15</td>
<td>18</td>
<td>16</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>16</td>
<td>14</td>
<td>9</td>
<td>8</td>
</tr>
<tr>
<td>25</td>
<td>15</td>
<td>13</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>30</td>
<td>12</td>
<td>9</td>
<td>6</td>
<td>5</td>
</tr>
</tbody>
</table>

**Figure 4.5. Time results comparison for percentage of the outliers vs. methods**

Figure 4.5 depicts the run time to be a function of the percentage of “outliers” in the data set every technique is expected to search for optimal results. From the figure 4.5 it concludes that the proposed EMPWC has lesser execution time results of 0.010 seconds in terms of percentage of outliers (0.5) which are 0.002
seconds, and 0.004 seconds lesser when compared to AMCEM and, ITB-SS methods respectively. The results of all the methods in terms of percentage of the outliers are discussed and tabulated in table 4.4.

**Table 4.4. Time results comparison for percentage of the outliers vs. methods**

<table>
<thead>
<tr>
<th>Percentage of outliers</th>
<th>Time (Seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ITB-SS</td>
</tr>
<tr>
<td><strong>0.1</strong></td>
<td>0.028</td>
</tr>
<tr>
<td><strong>0.2</strong></td>
<td>0.026</td>
</tr>
<tr>
<td><strong>0.3</strong></td>
<td>0.021</td>
</tr>
<tr>
<td><strong>0.4</strong></td>
<td>0.018</td>
</tr>
<tr>
<td><strong>0.5</strong></td>
<td>0.014</td>
</tr>
</tbody>
</table>

4.5.2. Outlier detection measurements

The Normalized Root Mean Square Error \( (NRMSE) \) is computed as,

\[
NRMSE = \frac{\sqrt{\text{Mean}[\{(y_{\text{guess}} - y_{\text{ans}})^2\}]}}{\text{std}[y_{\text{ans}}]} \tag{4.24}
\]

where \( y_{\text{guess}} \) and \( y_{\text{ans}} \) refer to vectors whose elements comprise the corresponding estimated values and the known answer values, for all of the data objects in the cluster \( s \). The mean and the standard deviation are computed over outlier data in the whole matrix.
In Figure 4.6 illustrates the results of the performance comparison of the NMSE for the existing techniques like ITB-SP ,ITB-SS, AMCEM and the new EMPWC algorithm. From the figure 4.6 it concludes that the NMSE value of the proposed EMPWC algorithm have lesser NMSE in comparison when compared to other existing techniques. From the figure 4.6 it concludes that the proposed EMPWC algorithm has lesser NMSE results of 6% which is 1%, 3%, and 5% lesser error value when compared to AMCEM, ITB-SS, and ITB-SP methods respectively.

**Detection Rate**

Correct detection rate is the number of outliers detected correctly by every technique.
Figure 4.7 depicts the performance comparison results of the outlier Detection Rate (DR) for the methods such as ITB-SP, ITB-SS, AMCEM and EMPWC in terms of threshold value of the Shannon entropy for each attribute. From the figure 4.7 it concludes that the proposed EMPWC has higher average detection rate of 82.45% which is 1.32%, 5.9%, and 7.81% higher when compared to AMCEM, ITB-SS, and ITB-SP methods respectively. The detection accuracy of this proposed EMPWC is higher when compared to existing methods, since the outlier values are optimized using PSO. This PSO algorithm exactly optimizes the outliers, thus increases clustering accuracy as well as reducing error rate. The results obtained of all the techniques with regard to detection rate are discussed and tabulated in table 4.5.
Table 4.5. Detection Rate(DR) vs. methods

<table>
<thead>
<tr>
<th>Threshold</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
<th>EMPWC</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>73</td>
<td>75</td>
<td>79</td>
<td>80.1</td>
</tr>
<tr>
<td>0.2</td>
<td>73.5</td>
<td>75.6</td>
<td>79.6</td>
<td>82</td>
</tr>
<tr>
<td>0.3</td>
<td>74.1</td>
<td>76.5</td>
<td>81</td>
<td>82.6</td>
</tr>
<tr>
<td>0.4</td>
<td>75.3</td>
<td>77</td>
<td>82</td>
<td>83</td>
</tr>
<tr>
<td>0.5</td>
<td>76.4</td>
<td>78</td>
<td>83</td>
<td>84</td>
</tr>
<tr>
<td>0.6</td>
<td>77.4</td>
<td>79</td>
<td>84.5</td>
<td>85.6</td>
</tr>
<tr>
<td><strong>Average DR</strong></td>
<td><strong>74.671429</strong></td>
<td><strong>76.585714</strong></td>
<td><strong>81.15714</strong></td>
<td><strong>82.48571</strong></td>
</tr>
</tbody>
</table>

Figure 4.8. False Alarm Rate(FAR) analysis vs. methods
In Figure 4.8 illustrates the performance comparison results of the False Alarm Rate (FAR) for the existing methods such as ITB-SP, ITB-SS, AMCEM and proposed EMPWC algorithm. From the figure 4.8 it concludes that the proposed EMPWC algorithm has lesser average FAR value of 17.33 % which is 1.41%, 5.73%, and 7.88% lesser value when compared to AMCEM, ITB-SS, and ITB-SP methods respectively. The results of all the methods in terms of FAR are discussed and tabulated in table 4.6.

**Table 4.6. False Alarm Rate (FAR) results vs. methods**

<table>
<thead>
<tr>
<th>Threshold</th>
<th>FIB</th>
<th>ITB-SP</th>
<th>ITB-SS</th>
<th>AMCEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>27</td>
<td>25</td>
<td>21</td>
<td>19</td>
</tr>
<tr>
<td>0.2</td>
<td>26.5</td>
<td>24.4</td>
<td>21</td>
<td>19</td>
</tr>
<tr>
<td>0.3</td>
<td>25.9</td>
<td>23</td>
<td>19</td>
<td>18</td>
</tr>
<tr>
<td>0.4</td>
<td>25.7</td>
<td>23</td>
<td>18</td>
<td>17</td>
</tr>
<tr>
<td>0.5</td>
<td>23.6</td>
<td>22</td>
<td>17</td>
<td>16</td>
</tr>
<tr>
<td>0.6</td>
<td>22.6</td>
<td>21</td>
<td>16.5</td>
<td>15</td>
</tr>
<tr>
<td>Average</td>
<td>25.216667</td>
<td>23.066667</td>
<td>18.75</td>
<td>17.33333</td>
</tr>
</tbody>
</table>

**4.6. SUMMARY**

The effectiveness of proposed EMPWC outlier detection method requires attribute frequency based results from weighted entropy function. Here the weighted entropy function considers both the Shannon and Jensen-Shannon Divergence (JSD) for measuring the likelihood of outlier candidates. The first step seems to be a pretty straight generalization of standardized EM for Gaussian mixtures; the second one is performed based on the results of the weight computation. Moreover in this research
work, the range values of outliers (o) get optimized employing the Particle Swarm Optimization (PSO). On the basis of this PSO technique, the data clustering results are seen to increase and therefore the algorithm proves higher efficient than other methods.