Chapter 4
Exploiting Swarm Behaviour of Simple Agents for Clustering Web Users' Session Data

Shafiq Alam, Gillian Dobbie, and Patricia Riddle

Abstract In recent years the integration and interaction of data mining and multi agent system (MAS) has become a popular approach for tackling the problem of distributed data mining. The use of intelligent optimization techniques in the form of MAS has been demonstrated to be beneficial for the performance of complex, real time, and costly data mining processes. Web session clustering, a sub domain of Web mining is one such problem, tackling the information comprehension problem of the exponentially growing World Wide Web (WWW) by grouping usage sessions on the basis of some similarity measure. In this chapter we present a novel web session clustering approach based on swarm intelligence (SI), a simple agent oriented approach based on communication and cooperation between agents. SI exploits the collective behaviour of simple agents, cooperation between the agents, and emergence on a feasible solution on the basis of their social and cognitive learning capabilities exhibited in the form of MAS. We describe the technique for web session clustering and demonstrate that our approach perform well against benchmark clustering techniques on benchmark session data.

4.1 Introduction

Data mining (DM) or Knowledge-Discovery and Data Mining (KDD), is the process of automatically searching large volumes of data for hidden, interesting, unknown and potentially useful patterns [1]. Data mining analyzes huge amounts of data for useful patterns using computational techniques from machine learning, information retrieval, computational intelligence and statistics [2]. With the rapid growth of web

Department of Computer Science, University of Auckland,
Private Bag 92019, Auckland, New Zealand
sala038@aucklanduni.ac.nz
{gill,pat}@cs.auckland.ac.nz

L. Cao (ed.), Data Mining and Multi-agent Integration, DOI: 10.1007/978-1-4419-0522-2_4,

data, web mining a sub domain of data mining has been introduced. Web mining tackles the information comprehension problem of the exponentially growing web data. Standard data mining techniques are applied to pre-process transform and extract patterns from web data. Web mining uses clustering, classification, association mining and prediction analysis to extract useful information from web documents. Web mining is further divided into web structure mining, web usage mining (WUM) and web content mining. We focus on WUM where the activities of more than 1.4 billion [1] internet users generate massive data and provide challenges for the automated discovery of interesting patterns among their usage behaviour. Organizations such as Google and Yahoo collect terabytes of data related to user activities, and analyze it for their business interests such as cross marketing, website organization, web site restructuring, recommender systems, web server performance improvement, and bandwidth management by caching and prefetching.
In recent years integration and interaction of data mining and multi agent system (MAS) has become a popular approach to tackle the problem of distributed data mining [3] [4]. The use of intelligent optimization techniques in the form of MAS has been shown to be beneficial for the performance of complex, real time, and costly data mining processes. Swarm Intelligence (SI) is one such paradigm that exploits the social and cognitive learning properties of vertebrates and insects, and models it through a multi agent system, with agents communicating with each other in a decentralized environment. The cooperative behaviour amongst the agents enables them to converge on an optimum solution. The two basic algorithms, ant colony optimization (ACO) and particle swarm optimization (PSO), have been found to be efficient in various domains of data mining. ACO is successfully implemented in classification, feature selection, rule mining and data clustering while the application of PSO can be found in data clustering, classification, pattern recognition, image processing, and recommender systems.
The main contributions of this chapter are:

- A description of web usage clustering in the context of a collective behaviour based multi agent environment
- A novel agent based technique for web session clustering based on PSO clustering, and a comparison of its performance with current techniques.

The rest of the chapter is organized as follows. Section 2 elaborates on the process of WUM and web session clustering approaches. Section 3 describes details of PSO and introduces the proposed PSO based clustering algorithm. Section 4 presents the pre-processing and clustering results. Section 5 overviews the related work in the area and section 6 introduces future work and concludes the paper.

[1] Internet usage statistics, the internet big picture http://www.internetworldstats.com/stats.htm

4.2 Web Usage Mining

Web usage mining (WUM) aims to discover interesting patterns among the fast increasing web users' activities on the WWW. It extracts hidden patterns in the visit sequence of the web users using standard data mining and KDD techniques. Web logs which record all the data related to the web users activities, needs to be passed through a sophisticated pre-processing stage. Web usage mining follows all the KDD steps; selection, pre-processing, pattern mining, post processing, and pattern analysis. Following are the main data mining techniques used to discover patterns in web usage data.

- Association rule mining
- Sequential pattern mining
- Classification rule mining
- Prediction analysis
- Clustering analysis

This section provides a detailed overview of web usage clustering practices.

4.2.1 Web Session Clustering

To understand the group behaviour of a particular class of users, an important step in web usage mining is to analyze the group behaviour of a user's sessions [5]. Clustering of web sessions is based on the data collected in the web server logs; gathered around on cache servers or in the cookies of client machines. Sometimes the process is backed by the structural and semantic information of the web pages. For web session clustering, primary attributes such as IP address, date time, page requested, page size, response and referrer are directly extracted from the web log while secondary attributes such as user visit, sessions, session length, episode, sequence of web usage and navigation, and semantic information are extracted by processing the primary attributes.

4.2.2 Session Identification

During a proper visit, web users follow a specific path related to their browsing behaviour and spend an arbitrary amount of time on each web page. The amount of time spent on a page is directly proportional to the interest of the user in that page. The sequence formed from such visits causes various hits on different pages. Such a sequence of visits is known as a web session. Identification of the session for a particular user can be by human intervention or automatic. Pseudo code of both techniques is given in Algorithm 1 and Algorithm 2 respectively. Both of these approaches have their own pros and cons. Each session must represent a single role

otherwise the clustering of web sessions will be biased and have a high risk of clustering web sessions which are totally unrelated.

Algorithm 1: Time Threshold based session identification

```
1  initialize sessionStartTime, logPointer, timeOutThreshold ;
2  while there exist (moreRecordsInLog) for a particular IP do
3      read(nextRecord);
4      if recordRequestTime-lastRequestTime ¡ timeOutThreshold then
5          append(currentSession, Record);
6      else
7          close (currentSession);
8          createNewSession(IP);
9      end
10 end
```

Algorithm 2: Behaviour shift based session identification

```
1  Initialize time=sessionStartTime, logPointer=1 ;
2  while there exist (moreRecordsInLog) for a particular IP do
3      read(nextRecord);
4      if ShiftInBrowsingBehaviour == True then
5          append(currentSession, Record);
6      else
7          close (currentSession);
8          createNewSession(IP);
9      end
10 end
```

4.2.3 Web Session Clustering Techniques

Web session clustering exploits the three main dimensions of web usage data; time dimension, semantic usage behaviour dimension and browsing sequence dimension. For time dimension based clustering, the Euclidean distance is used to measure the distance between two sessions. Each session is transformed to a data vector with finite attributes representing time dimension information of a session. The Euclidean distance measure calculates the distance between two session vectors and the clustering algorithm decides in which cluster the session is to be placed.

$$d(x,y) = \left(\sum_{i}^{n} (x_i - y_i)^2 \right)^{(1/2)} \tag{4.1}$$

where x_i is the i^{th} attribute value of data vector x and y_i is the i^{th} attribute value of data vector y. The dimension of each data vector is from 1 to n representing the attributes of a session.
The semantic clustering of a session involves semantic information in terms of page and topic similarities. The session can be clustered on the basis of the relatedness of the pages viewed by each user during their respective session. The similarity of pages is measured using term frequency inverse document frequency $(tf-idf)$ measure shown in equation 4.2.

$$w(i,j) = tf_{i,j} * log\left(\frac{N}{df_i}\right) \tag{4.2}$$

where $tf_{i,j}$ is the frequency of term i in document j, df_i is the number of documents possessing term i, and N is the total number of documents. Some approaches perform generalization of sessions to increase the semantic coverage of the session [2]. The method in [5] first generalized the session in attribute-oriented induction according to a data structure, called page hierarchy-partial ordering of the Web pages, and then clustered using BIRCH. In [6] click stream clustering is performed using Weighted Longest Common Subsequences (WLCS).
The area which is mostly investigated by researchers is the browsing sequence based session clustering. The sequence of each session consists of a page-hit hierarchy of the web user and forms a labeled edge graph. The distance between these graphs are then calculated for clustering the related pages into identical clusters. The Levenshtein distance method gives edit distance between two sequences of navigations.

$$dG(G1,G2) = 1 - 2\left[\frac{L(S1,S2)}{(\|E(G1)\| + \|E(G2)\|)}\right] \tag{4.3}$$

where $L(S1,S2)$ is the Levenshtein distance between path $S1$ and path $S2$ and $\|E(G)\|$ shows the number of nodes in the graph. Some approaches combine the time dimension with browsing sequence to identify the relative importance of a visit. Weighted longest common subsequence (LCS) is the common subsequence among any two sessions. WLCS creates a sequence which considers the similarity of the common region weighted by time and importance of that region. The *similarity* component shows how similar the two paths are and *importance* shows how important the region of intersection is in terms of time spent on that region [6]. *Similarity* of two sequences is measured by

$$S_i = \frac{Min(Seq1_i, Seq2_i)}{Max(Seq1_i, Seq2_i)} \tag{4.4}$$

where S_i is the similarity of the i^{th} visits of both sessions and its value is from 0 to 1. The average similarity of the two sequences is

$$S = \frac{1}{L}\sum_{i}^{L}\left(\frac{Min(Seq1_i, Seq2_i)}{Max(Seq1_i, Seq2_i)}\right) \tag{4.5}$$

where, L is the length of the longest common sub sequence. The *importance* component is calculated as

$$imp = \left[\frac{TimeOfLCS1}{TotalTimeOfSeq1}\right] \times \left[\frac{Min(Seq1_i, Seq2_i)}{Max(Seq1_i, Seq2_i)}\right] \tag{4.6}$$

Once the similarity values are calculated then the next step is to create a similarity graph of the users where similar activities are automatically grouped into identical clusters. For generalized conceptual graph clustering, the author [7] added pages similarity for clustering web sessions. In domain taxonomy based clustering, the similarities of two pages are calculated as.

$$S(c1,c2) = \frac{2 \times [depth(LCA(c1,c2))]}{(depth(c1) + depth(c2))} \tag{4.7}$$

where LCA is the lowest common ancestor of concepts c1 and c2. The equation is based on Generalized Vector-Space Model [8], where $c1$ and $c2$ are concepts in the hierarchy, and $depth$ is the number of edges from the concept to the top of the hierarchy.

4.3 Swarm Intelligence

Swarm Intelligence (SI), inspired by the biological behaviour of animals is an innovative distributed intelligence paradigm for solving optimization problems [9]. It is a state of the art optimization technique based on the communication and cooperation of autonomous agents in a multi agent environment. The two main swarm intelligence algorithms; Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) are widely used for optimization of discrete and continuous problems. Both of the techniques have been successfully used for the solution of different optimization problems such as NP hard problems, data mining, distributed systems, power systems, hybrid systems, and complex systems. In this section we discuss the details of PSO.

4.3.1 Particle Swarm Optimization

PSO is an optimization technique originally proposed by [10] and is based on the inspiration from the swarm behavior of birds, fish and bees when they search for food or communicate with each other. The particles or birds correspond to agents, the swarm a collection of particles, represents a multi agent system and the swarming behavior in the particles is like agent communication [11][12][13]. For PSO, the solution space of a problem is represented as a collection of agents where each agent represents an individual solution and the MAS represents the solution space for a particular iteration. In PSO the agents are initialized randomly to a solution

set from the solution space. The velocity of the agents causes change in the agents position. The agents maintain their current velocity value and their personal best position (pBest) while moving from one position to another. The pBest maintained by every agent is the best ever position (fitness) found by that agent. The swarm also maintains a best value which is called global best position (gBest). The gBest value is the position representing best fitness value achieved for all agents of the MAS. The pBest value is calculated by equation 4.8.

$$pBest_i(t+1) = \begin{cases} pBest_i(t) & \text{if } f(X_i(t+1)) \text{ isNotBetterThan } f(pBest_i(t)) \\ X_i(t+1) & \text{if } f(X_i(t+1)) \text{ isBetterThan } f(pBest_i(t)) \end{cases} \tag{4.8}$$

where $X_i(t+1)$ is the current position of the agent, $pBest_i(t)$ is the personal best position and $pBest_i(t+1)$ is the new best position. After finding the new personal best position, the next step is to calculate the global best position, which can be extracted by $gBest(t) = argmin_{i=0}^{n} pBest_i(t)$, where i is the index of each agent ranging from 0 to the total number of agents n. The velocity of each agent is influenced by two learning components: the cognitive component $(pBest - X_i(t))$ and the social component $(gBest - X_i(t))$. The cognitive component represents learning from history and experience while the social component represents learning from the other fellow agents of the MAS. The cognitive and the social component guide the agent towards the best solution. The velocity update equation guided by the cognitive and social learning component is shown in equation 4.9.

$$V_i(t+1) = \omega * V_i(t) + q1r1(pBest - X_i(t)) + q2r2(gBest - X_i(t)) \tag{4.9}$$

where $V_i(t)$ is the current velocity, $V_i(t+1)$ is the new velocity, ω is the inertia weight, $q1$ and $q2$ are the values which weigh the cognitive and social components and $r1$ and $r2$ are two randomly generated numbers ranging from 0 to 1. The range for the velocities of the agents is from $-V_{max}$ to V_{max} . Position of the agent is updated using the position updating equation 4.10.

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{4.10}$$

After calculating the new position of each agent, the swarm looks for the global fitness which is evaluated for the final solution during a particular iteration. If the solution doesn't fulfill the specified criteria, the next generation of the swarm is iterated. The process continues until the stopping criteria i.e. maximum number of iterations or the minimum error requirement, is fulfilled. The number of agents in the system is selected according to the problem complexity. Algorithm 3 shows the pseudo code of the PSO process.

Algorithm 3: Particle Swarm Optimization

```
foreach particle do
    initialize all parameters;
end
repeat
    foreach particle do
        calculate fitness value;
        if fitness value is better than pBest then
            set current value to pBest using (8);
        end
        choose the particle with the best fitness value;
    end
    foreach particle do
        calculate particle velocity using (9);
        update particle position using (10);
    end
until stopping criteria unfulfilled ;
```

4.3.2 PSO Based Web Usage Clustering

The approach we propose in this paper cluster on time and browsing sequence dimensions of the web usage data set. We formulated sessions as particles for the particle swarm optimization algorithm using the idea of Cohen et al. [14]. The formulation of the problem for this approach is discussed in the following paragraph. We consider swarm as a multi agent system and an individual particle as an agent of the MAS. Each session vector contains attributes of a user session i.e. session length, number of pages visited during that session, and amount of data downloaded etc. All the sessions recorded for user activities represent the input data space for the clustering problem. Each agent of the MAS is initialized randomly to one of the input session vectors. Once the initialization of the entire system is completed, the next step is to iterate each agent of the system to find suitable position. After completion of the first iteration each agent is evaluated for its performance i.e. personal best position using equation 4.8. This value effects the learning of the agent from its experience. The agent uses personal best position to influence its velocity. The cognitive component is $q1r1(pBest - X_i(t))$ where $q1$ and $r1$ are the two constants which weight the cognitive component. To learn from the experience of the whole swarm, the agent takes its inspiration from the global best position called *gBest* position. To obtain the *gBest* value, the swarm evaluates each agent and selects the best single position/ fitness of all the particles and sets this value as *gBest* for the current iteration of the swarm. The social learning component $q2r2(gBest - X_i(t))$ causes the movement of the agent to be influenced from the experience of the entire swarm, $q2$ and $r2$ are the weighing constants of the social component. The self organizing component of the agents $q3r3(X_i(t) - Y_i(t))$ influence the particle movement towards the best position in its sub population. $Y_i(t)$ is the position of the session vector of a particular cluster. The social learning component, the cognitive learning

component and the self organizing component decides the movement direction of the agent.

$$V_i(t+1) = \omega * V_i(t) + q1r1(pBest - X_i(t)) + q2r2(gBest - X_i(t)) + q3r3(X_i(t) - Y_i(t)) \tag{4.11}$$

For the solution of the clustering problem where the agent does not take its inspiration directly from the experience of the entire swarm and its movement is guided by the cognitive component and self organizing component only, the value of *gBest* is ignored i.e. $q2r2(gBest - X_i(t)) = 0$. In such cases a single agent represents one clustering centroid instead of a complete clustering solution. Equation 4.11 calculates the velocity of the agent which is then added to the current position value to find the new position of the agent.

$$X_i(t+1) = X_i(t) + V_i(t+1) \tag{4.12}$$

while $X_i(t+1)$ is the new position of the agent, $X_i(t)$ is the current position of the agent and $V_i(t+1)$ is the current velocity of the particle. The agents change their position with respect to their sub population while the sub population i.e. the session vector, do not change their position.
In our experiment, each agent consists of session attributes; total time of a session, number of pages visited, and amount of data downloaded in a particular session. Each agent represents a part of the clustering solution as a centroid of that cluster, while the entire swarm represents a solution to the clustering problem. Following are the main attributes of an agent, which it keeps while moving through the solution space.

- *ParticleId:* it uniquely identifies an agent or a centroid session.
- *DistanceFromEachSession:* an array which represents the distance of the agent to each session at a particular iteration. We used the Euclidean distance measure in our experiment. The closest sessions to the agent are won by that agent and added to the *WonSessionVectors*.
- *WonSessionVectors:* an array which represents the session vectors won by an agent at a given iteration. The agent organizes itself among the current won sessions. This causes the agent to learn from the neighborhood and organize itself within its sub population.
- *SessionAttributeValues:* represents the current values of the agent in each dimension in the form of a data vector. The more session attributes the easier it is to find the similarities among sessions.
- *PBest:* is the position of the nearest session to the agent achieved so far. This is obtained by keeping track of the position of the best previous session.

Once the initialization is completed, the agents are now moved from their initial position (starting session), guided by the social, cognitive and self organizing component. The cognitive component of the algorithm is encoded as $(pBest - X_i(t))$.

The social component is encoded as $(gBest - X_i(t))$ and a self organizing term as $(Y_i(t) - X_i(t))$ where $Y_i(t)$ is the current position value of the agent, *pBest* is the personal best position found by the agent so far and *gBest* is the global best position, however in this particular case the value of the social term is not as important because the agent should not follow the whole swarm but only its given sub population. So we ignore the $(gBest - X_i(t))$ for clustering the web usage session as discussed earlier. The self organizing term is more important as it causes a change in the velocity of an agent towards the current session attribute. After each iteration, the swarm changes its position by winning the nearest sessions, recalculates all its parameters, organizes itself according to the new session vector won by each agent. The process continues until there is no significant change in the position of the agents or the number of maximum iterations is approached or no movement of data vectors from one cluster to another cluster is observed.

4.4 Experimental Results

In this section, we explain the preprocessing and clustering results respectively.

4.4.1 Data, Pre-processing and Usage Statistics

For experiments and performance evaluation of our approach we chose the NASA web log file[2], which contains HTTP requests to NASA Kennedy Space Centre's web server from 1^{st} July 1995 to 28^{th} July 1995. There are 1891715 requests in the log and the log size is 195 MB in text format. We analyzed the logs containing one day of HTTP requests dated 1st July 1995. The log was first passed through all the pre-processing steps, data cleaning, structuring and summarization. The details of the results after the pre-processing step are given in Table 4.1. The results reveal

Table 4.1: Results of the pre-processing

Total Requests	**64578**	**Successful Requests**	**23795**
Pure Requests	25387	users having $<$ 10 requests	4591
Distinct pages requested	1096	user having $>$ 10 requests	408
Unsuccessful requests	1592	Average Request per user	13
Distinct user	4999	Total requests $>$ 10 request per user	10121
Images request	61%		

Total sessions	**815**
Session $>$ 10 **request**	432
Average request per session	13
Average session per user	2

[2] http://ita.ee.lbl.gov/html/contrib/NASA-HTTP.html

the fact that more than 60% of the web requests recorded in the web logs are image requests and are useless in the context of web session clustering. The importance of the pre-processing stage is verified by the ratio of successfully selected requests to the total requests i.e. $1/3$ of the total request are selected for analysis.
Fig. 4.1 summarizes the usage statistics of visits, users, session and responses generated by the web server. Fig. 4.1 (a) shows the number of pages against the number of users, after the pre-processing phase. The distribution shows that most of the users are from the class where the number of pages viewed is from 5 to 15. The number of users decreases gradually with an increase in the number of pages viewed. In Fig. 4.1 (b) the number of requests is plotted against 30 minute time interval starting from July 1, 1995 12:00:01 AM to 12:00:00 PM, Fig. 4.1 (c) elaborates the number of page requested against the session length. The distribution shows that most of the sessions have an iteration number less than 5, which are known not to be representative of the real usage behavior, so they are ignored. Fig. 4.1 (d) shows the session count and percentage against time intervals. The time interval of 11 to 20 minutes gets the highest session count, which demonstrates that most of the web users have a session time between 10 to 30 minutes and longer sessions are rare in web logs.

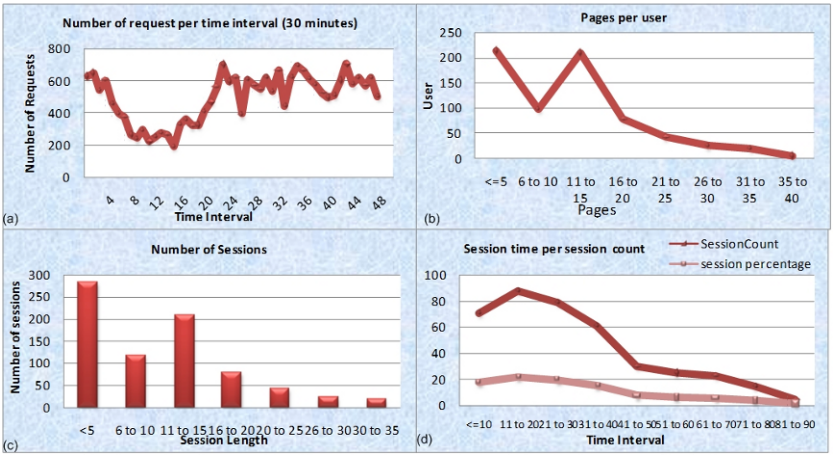

Fig. 4.1: (a) Time and request per 30 minutes distribution (b) Page viewed and number of user distribution (c) Session-number of request distribution (d) Session-time distribution

4.4.2 Clustering Results

After pre-processing and removal of sessions with < 10 requests, 432 sessions were selected for our clustering analysis. The purpose of the experiment was to group the session on the basis of the session attribute values using the agent based particle swarm optimization clustering approach for comparison of our approach with

K-means. Fig. 4.2 shows grouping of the sessions in 5, 10, 15 and 20 clusters respectively. Most of the users have similar behavior and can be grouped in 2-4 active clusters. Sessions with higher amounts of data downloaded, high visit times and larger

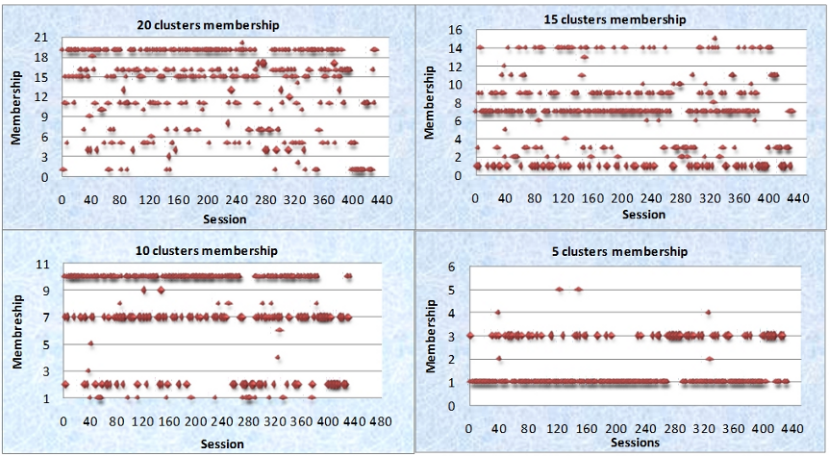

Fig. 4.2: (a) 20 Cluster Membership (b)15 Cluster Membership (c) 10 Cluster membership (c) 5 Cluster membership

iteration number were grouped into outliers. For simplification and comparison we divided the dataset into four sub datasets each with 100 user sessions. The visualization of the relationship of different session values and their clustering membership of the web sessions is shown in Fig. 4.3, which verify that similar web user sessions fall in the same clusters. Table 4.2 shows the performance of PSO clustering in terms of cluster distribution and the intra cluster distance. Taking into account the density of clusters, uniform initialization, and agent's convergence nature are some of the additional advantages. The overall fitness of the PSO is better than K-means clustering. For comparison purposes, we initialized the centroids of both the algorithms i.e. the K-means and PSO clustering to the same values. We performed a variety of experiments with initialization, parameter selection and iterations to verify the efficiency of the approach. For the PSO-clustering algorithm the parameters were set to the range $Vmax = [0.1, 0.04]$, $q1 = [.01, 0.9]$, $q2 = 0$, $q3 = [0.01, 0.9]$, $\omega = [0.01, 0.09]$, acheiving the results shown in Table 4.2. The number of iterations on which the solution is obtained in most of the cases was below 100. To access the time consistency of the approach, we scaled the number of iteration to a maximum iteration of 1000, however we have not found any inconsistency or abrupt change in execution time. The relationship between the number of iterations of PSO clustering and execution time was observed as linear as demonstrated in Fig. 4.4.

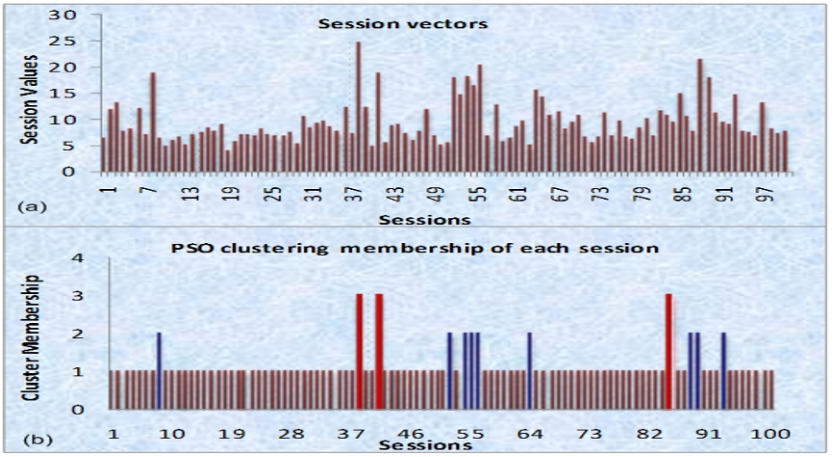

Fig. 4.3: (a) Session data (b) PSO clustering members

Table 4.2: Comparison of K-means and PSO clustering

Log	K-means		PSO	
	Mean IntraCluster Dist.	Fitness	Mean IntraCluster Dist.	Fitness
1	81.8211	245.463	81.4863	**244.459**
2	35.0334	105.1002	34.85	**104.55**
3	27.7769	55.5538	25.8365	**51.6731**
4	59.1294	118.2588	59.1367	118.2734

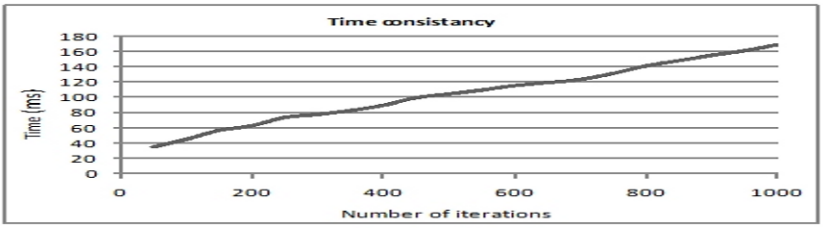

Fig. 4.4: Execution time and number of iteration

4.5 Related Work

PSO was introduced for data clustering by Van der Merwe and Engelbrecht [15] by initializing randomly created particles to a vector containing centroids of the clusters. The evaluation of the method was based on the cost function that evaluates each candidate solution based on the proposed cluster's centroids. In [16], the authors applied PSO with Self-Organizing Maps (SOM) where SOM are used for grouping and PSO optimizes the weights of the SOM. Chen and Ye [17] represented each particle

corresponded to a vector containing the centroids of the clusters. The results were compared with k-means and fuzzy c-mean using the objective function based on intra-cluster distance. Omran et al. [18] proposed a dynamic clustering algorithm using PSO and k-means for image segmentation, which finds the number of clusters in the data automatically by initially partitioning the data set into a large number of clusters. Cohen et al. [14] used PSO for data clustering where each particle represents a portion of the solution instead of entire clustering solution. In [19] the authors proposed a generation based evolutionary clustering technique which uses the concept of consumption of weaker particles by strong particles. The approach provided a solution for the clustering problem on different levels of compactness.
In the web usage domain the ACO algorithm was used by [20] for clustering based prediction of web traffic. AntClust was introduced for web session clustering by [21] and in [22], the authors proposed ant-based clustering using fuzzy logic. In the web usage domain PSO is implemented [23], which combines improved velocity PSO with k-means to cluster web sessions. In [24] the author proposed particle swarm optimization approach for the clustering of web sessions.
Recently some of the research [3] [4] and [13] have focused on data mining with multiagent integration and interaction.

4.6 Conclusion and Future Work

Integration and interaction of data mining with multi agent system is beneficial for mining the distributed nature of WWW. Web session clustering, one of the important WUM technique, aims to group similar web usage sessions into identical clusters. We clustered the pre-processed WUM data using a swarm intelligence based optimization, PSO based clustering algorithm. In the proposed approach, simple agents communicate with each other and cooperate and produce the solution to the clustering problem. Each agent represent a single cluster and the swarm of agents represent the complete clustering solution. We showed the performance of the algorithm is better than K-means clustering. The future directions in the area are the integration of different parameters for clustering, development of accurate similarity measures, PSO parameter automation and involvement of optimization algorithms in other areas of web usage mining.

References

1. Frawley, W., Piatetsky-Shapiro, G., Matheus, C. : Knowledge Discovery in Databases: An Overview. AI Magazine: pp. 213-228, (1992)
2. Edelstein, H.A.: Introduction to data mining and knowledge discovery (3rd ed). Two Crows Corp, Potomac, MD, (1999)
3. Cao, L., Gorodetski, V.: AREA OVERVIEW-Agent & data mining interaction (ADMI). In: WI-IAT 2006 IADM Workshop panel discussion, Hongkong (2006)

4. Cao, L., Luo, C., Zhang, C.: Agent-Mining Interaction: An Emerging Area, AIS-ADM07, LNAI 4476, 60-73, Springer, (2007)
5. Fu, Y., Sandhu K., Shih, M-Y.: A Generalization-Based Approach to Clustering of Web Usage Sessions, Revised Papers from the International Workshop on Web Usage Analysis and User Profiling, p.21-38, (1999)
6. Banerjee, A., Ghosh, J.: Clickstream Clustering using Weighted Longest Common Subsequence. In: Proceedings of the 1st SIAM International Conference on Data Mining: Workshop on Web Mining. (2001)
7. Nichele, C. M., Becker, K.: Clustering Web Sessions by Levels of Page Similarity. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD '06): 346-350. (2006).
8. Ganesan, P., Garcia-Molina, H.,Widom, J.: Exploiting Hierarchical Domain Structure to Compute Similarity. ACM Transactions on Information Systems (TOIS), v.21, n.1, 64-93. (2003)
9. Abraham, A., Guo, H., Liu, H.: Swarm Intelligence: Foundations, Perspectives and Applications. Swarm Intelligent Systems, Studies in Computational Intelligence, (eds.), pp. 3-25. Springer,(2006)
10. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. International Conference on Neural Networks (ICNN '95), Vol. IV, Perth, Australia (1995)
11. Engelbrecht, A.P.: Fundamentals of Computational Swarm Intelligence. John Wiley and Sons, (2005)
12. Kennedy, J., Eberhart, R.C.: Swarm intelligence. Morgan Kaufmann Publishers, (2001)
13. Altshuler, Y., Bruckstein, A.M.,Wagner, I.A.: On Swarm Optimality In Dynamic And Symmetric Environments. In: Second International Conference on Informatics in Control, Automation and Robotics (ICINCO), Barcelona, Spain, (2005)
14. Cohen, S.C. M., de Castro, L. N.: Data Clustering with Particle Swarms. In: IEEE Congress on Evolutionary Computations, Vancouver, BC, Canada, (2006)
15. Van der Merwe, D.W, Engelbrecht, A.P.: Data clustering using particle swarm optimization. In: Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), Canbella, Australia. pp. 215-220, (2003)
16. Xiao, X., Dow, E.R., Eberhart, R.C., Ben Miled, Z., Oppelt, R. J.: Gene clustering using self-organizing maps and particle swarm optimization. In: Proceedings of Second IEEE International Workshop on High Performance Computational Biology, Nice, France, (2003)
17. Chen, C.-Y., Ye, F.: Particle swarm optimization algorithm and its application to clustering analysis. In: Proceedings of IEEE International Conference on Networking, Sensing and Control. pp. 789-794, (2004)
18. Omran, M. G. H., Salman, A., Engelbrecht, A. P.,: Dynamic Clustering Using Particle Swarm Optimization with Application in Image Segmentation. Pattern Analysis and Applications, Vol. 8, pp. 2-344,(2005)
19. Alam, S., Dobbie, G., Riddle, P.: An Evolutionary Particle Swarm Optimization Algorithm For Data Clustering. In: Proceedings of IEEE International Swarm Intelligence Symposium. Missouri, USA ,(2008)
20. Abraham, A.: Natural Computation for Business Intelligence from Web Usage Mining. In: Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC'05), pp. 3-10,(2005)
21. Labroche, N., Monarch, N., Venturini, G.: AntClust: Ant clustering and web usage mining. Genetic and Evolutionary Computation. Chicago, IL. Lecture Notes in Computer Science 2723. Berlin, Heidelberg, Germany: Springer-Verlag. pp 25-36,(2003)
22. Steven Schockaert , Martine De Cock, Chris Cornelis, Etienne E. Kerre.: Clustering web search results using fuzzy ants. International Journal of Intelligent Systems, Volume 22, Issue 5, Pages 455 - 474,(2007)
23. Chen, J. Z., Huiying : Research on Application of Clustering Algorithm Based on PSO for the Web Usage Pattern. Wireless Communications, Networking and Mobile Computing, (2007)
24. Alam, S., Dobbie, G., Riddle, P.: Particle Swarm Optimization Based Clustering of Web Usage Data. In: ACM/WIC/IEEE International conference on web intelligence. Sydney, Australia,(2008)