Role Based Particle Swarm Optimization Algorithm To Cluster Web Documents

Ingyu Lee
Sorrell College of Business
Troy University
inlee@troy.edu

Byung-Won On
School of Information Systems
Singapore Management University
bwon@smu.edu.sg

Abstract

Web documents clustering becomes an essential technology with the popularity of the Internet. The main goal of the document clustering is to minimize the distance within the same group while maximizing the distance between different groups. Since the dimension is very large in document clustering, particle swarm optimization (PSO) algorithms have been used to find optimal clustering. However, particle swarm optimization algorithms generally require a lot of iterations or sometimes fails to converge global minimum in case of the original corpus is extremely skewed. In this paper, we are proposing a role based particle swarm optimization (RoleBasedPSO) algorithm which assigns different roles to different groups of particles. Experimental results show that RoleBasedPSO shows 5% better performance in terms of accuracy and requires less number of iterations to converge to global minimum.

Keywords: particle swarm optimization, data mining, document clustering, entity resolution

1 Introduction

Document clustering is a fundamental operation used in unsupervised document organization, automatic topic extraction, and information retrieval. Especially, document clustering is getting popular with the advent of Internet web documents. The main goal of document clustering is to minimize the distance within the same group (compactness) while maximizing the distance between different groups (betweenness) as shown in Figure 1 [1]. Assume we have N documents, $d_1, \ldots, d_N$, and k clusters, $C_1, \ldots, C_k$, in the corpus. Then, document clustering is defined as an optimization problem with an objective function which is defined as

$$\min \frac{\sum_{k=1}^{N} \sum_{d_i \in C_k} (d_i - m_k)^2 / |C_k|}{\sum_{i,j \in C} (m_i - m_j)^2} \quad (1)$$

where $|C_k|$ is the number of documents in cluster C_k, and m_i is the centroid for cluster C_i [2, 3, 4]. Intuitively, the equation is to find minimum value of the ratio between intra cluster distance and inter cluster distance. A set of cluster which compacts within the same cluster and spreads between the different clusters minimizes the equation.

On the other hand, particle swarm optimizations, inspired by bird flocks, fish schools and swarms of insects, have been used to solve an optimization problem with large dimensional space [5, 6, 7, 8]. A PSO algorithm generally has three major phases: evaluation, comparison and imitation. Each particle evaluates itself using a fitness function, and then compares with the values of other particles. Finally, each particle imitates itself to reach the best result particle. At each iteration, swarm of particles are flying through the parameter space and searching for optimum fitness function value. Each particle is characterized by position vector $x(t)$ and velocity vector $v(t)$. Each particle has individual knowledge of $pbest_i$ which is its own best-so-far position and social knowledge $gbest$ which is the $pbest$ of its best neighbor. At each time step, the velocity and position are updated as

$$\begin{aligned} v_i(t+1) &= wv_i(t) \\ &+ \alpha U(0,\psi_1)(pbest_i - x_i(t)) \\ &+ \beta U(0,\psi_2)(gbest - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (2)$$

where w is an inertia weight, α and β are weight factors for local and global search, respectively. With a large inertia weight, algorithm explores for global optimal but it could fluctuate than converge. On the other hand with a small inertia weight, algorithm might converge on a local minimal rather than a global optimum value.

Sometimes, particle swarm optimization algorithms converge on local minimum since all the particles search within local area. To overcome this problem, in this paper, we are proposing a role based particle swarm optimization algorithm (RoleBasedPSO). In a role based particle swarm optimization algorithm, we divide the particles into three

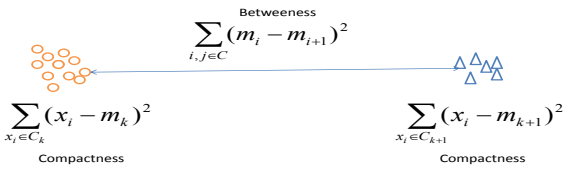

Figure 1. Document Clustering: Compactness and Betweenness: Goal is to minimize betweenness and maximize compactness.

different groups and each group of particles has a different role. One group of particles are responsible to search the global space with a relatively large inertia weight, the second group of particles stabilizes algorithm to converge with a relatively smaller inertia weight. The third group of particles change the inertia weight value linearly as in assumption that particles are reaching to global minimum with the number of iterations. We conjecture that dividing particle groups into roles prevents the algorithm converges on local minimum.

This paper consists of the followings. Section 2 describes about a role based particle swarm optimization algorithm. Experimental results on web documents are following in Section 3. Section 4 describes about related works. Concluding remarks and future plans are described in Section 5.

2 Role Based Particle Swarm Optimization

In traditional particle swarm optimization algorithm, every particle moves to the current global best position with the same speed as in Equation 2 (*i.e. all particles imitate the same global best particle*. However, this could lead to local minimum rather than searching for global minimum value. To prevent the algorithm converges to local minimum, we are using different schemes to update the velocity of each particle. We divided the particles into several groups and each group of particles are using different formula to update the position. We call this a role based particle swarm optimization (RoleBasedPSO) algorithm. Figure 2 shows a role based particle swarm optimization algorithm with three different groups. One group of particles are searching for a global minimum value by using a bigger inertia weight value w. The second group of particles are using a linearly decreasing inertia weight value as does in a traditional particle swarm optimization algorithm. The third group of particles are using a smaller inertia weight value to thoroughly search local area.

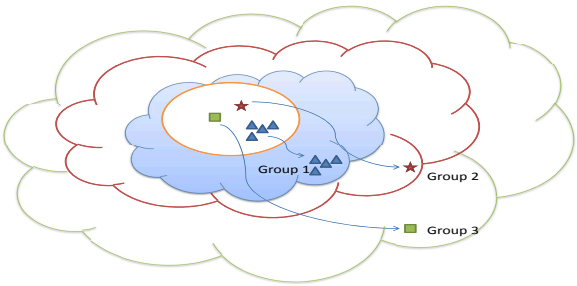

Figure 2. RoleBased PSO: Different groups explore different domain spaces. Group 1 explores locally, group 2 explore the mid range and group 3 search globally.

RoleBasedPSO()
% *pbestValue* indicates local best fitness value
% *pbest* indicates local best particle
% *gbestValue* indicates global best fitness value
% *gbest* indicates global best particle
Initialize the particles, p, as centroid from solution space
while max iteration is not reached or not converged **do**
  for each particle p_i **do**
    Compute the fitness function, $f(p_i)$
    if $f(p_i)$ is smaller than the $pbestValue_i$, local best **then**
      Change the local best fitness value, $pbestValue_i$ with $f(p_i)$
      Change the local best solution, $pbest_i$ with p_i
    else
      Use a proper equation to update the position based on the role of each particle, p_i

$$\begin{aligned} w_{group1} &= 0.72 \\ w_{group2} &= w - (iter-1) * 0.01 * w \\ w_{group3} &= 0.72/exp((iter-1)/maxit) \end{aligned} \quad (3)$$

      Change the position

$$\begin{aligned} v_i(t+1) &= wv_i(t) \\ &+ \alpha U(0,1)(pbest_i - x_i(t)) \\ &+ \beta U(0,1)(gbest - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (4)$$

    end if
  end for
  Find the global best fitness value among $pbestValue$ and set the value as $gbestValue$ and the particle as $gbest$
end while

Figure 3. Role Based Particle Swarm Optimization Algorithm: Three different groups of particles search different areas of domain. Each groups play different roles to find global minimum.

Therefore, Equation 2 is modified as

$$\begin{aligned} v_i(t+1) &= w^* v_i(t) \\ &+ \alpha U(0,\psi_1)(pbest_i - x_i(t)) \\ &+ \beta U(0,\psi_2)(gbest - x_i(t)) \\ x_i(t+1) &= x_i(t) + v_i(t+1) \end{aligned} \quad (5)$$

where w^* depends on particle groups. For particles of a group which search globally, the weight(w^*) is a fixed value (e.g. 0.72). For particles of a group which search locally, the weight (w^*) linearly decreases according to an iteration (e.g. decreasing 1% per iteration with $w - (iter - 1) * 0.01 * w$). For the last group, weight (w^*) changes exponentially (e.g. as in $0.72/exp((iter-1)/maxit)$). The details of the algorithm is shown in Figure 3. Considering each group of particles are searching for different areas of domain, we conjecture that the algorithm avoids local minimum and could reach a global minimum value.

3 Experiment Results

Data Sets: To show the performance of RoleBasedPSO algorithm, we choose web document name data set: *Bekkerman* [9] name data set, *Wikipedia* [10] and *ECDL* person name [10] as shown in Table 1. *Bekkerman* name data set are the results of google search top 100 pages, and *Wikipedia* and *ECDL* is the results of $Yahoo$ search top 100 pages. Some name data set for *Wikipedia* and *ECDL* allows multiple memberships (*i.e. one document may belong to multiple clusters*). However, we manually assigned to only one cluster to simplify the problem.

Web document name data set are extremely skewed in terms of cluster size and number of clusters as shown in Figure 4. Furthermore, one or two document members in the corpus dominate the data sets. For example, *Cheyer* and *Kaebling* from *Bekkerman* name data set has two cluster categories. All documents belong to the first category except one document which belongs to the second category. Therefore, conventional *Kmeans* algorithm has a difficulty to find a proper cluster since the algorithm relies on random initial points [11]. Contrarily, a PSO algorithm is relatively efficient in large dimensional space and relatively stable than relying on randomly choosen centroids.

Creating Document-Document Graph G: To create a terminology document matrix A, we used TMG [12] software package with spamming, and dropped common words using dictionary, and then applied normalization. Each A_{ij} element indicates the term frequency (TF) multiplied by inverse document frequency (IDF) of terminology t_i in the document d_j as in $A_{ij} = TF_{ij} * IDF_{ij}$. Then, we generate document-document matrix G by multiplying document term matrix A^T with term document matrix A as in $G = A^T \times A$. The $G(i,j)$ element in the matrix indicates the similarity value of two documents d_i and d_j. In other words, G_{ij} element holds sum of multiplications of d_{ik} and d_{jk} values for each terminology t_k as in

$$G = A^T \times A(i,j) = \sum_{k \in \{Term\}} d_{ik} \times d_{jk} \quad (6)$$

where $\{Term\}$ is a set of terminology. Intuitively, if two document d_i and d_j have a lot of common terminology, then the similarity value of two documents is large.

After we generate a document-document matrix G, we applied a PSO algorithm to cluster documents. We used 0.72 for the starting inertia weight value, and 1.49 for α and β in the algorithm according to [13, 14, 15] with a fitness function as in

$$\frac{\sum_{k=1}^{N} \sum_{x_i \in C_k} (x_i - m_k)^2 / |C_k|}{\sum_{i,j \in C} (m_i - m_j)^2} \quad (7)$$

where m_k is a centroid for cluster C_k, and $|C_k|$ is the number of members in cluster C_k. At each iteration, we evaluate

	Name	Pages	Classes
Bekkerman	Adam Cheyer	97	2
	William Cohen	88	10
	Steve Hardt	81	6
	David Israel	92	19
	Leslie Pack Kaebling	89	2
	Bill Mark	94	8
	Andrew McCallum	94	16
	Tom Mitchell	92	37
	David Mulford	94	13
	Andrew Ng	87	31
	Fernando Pereira	88	19
	Lynn Voss	89	26
ECDL	Allan Hanbury	68	2
	Andrew Powell	52	19
	Anita Coleman	72	9
	Christine Borgman	89	9
	Donna Harman	94	7
	Edward Fox	64	16
	Gregory Crane	83	4
	Jane Hunter	41	15
	Paul Clough	65	14
Wikipedia	John Jennedy	94	27
	George Clinton	94	27
	Paul Collins	94	37
	Michael Howard	92	32
	Tony Abbott	91	7
	David Lodge	91	11
	Alexander Macomb	86	21

Table 1. Name Data Set Statistics: Each name has different number of documents and different number of categories which makes clustering to be more challenging.

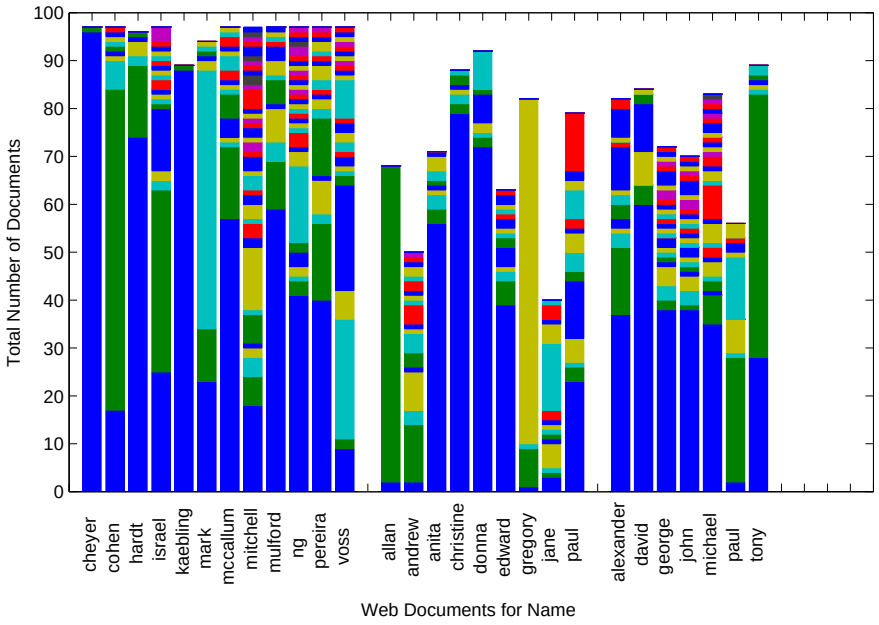

Figure 4. Statistics of Problem Set: Each color represents different clusters and height shows the number of documents. Name data sets are extremely skewed and one or two classes dominate the whole document set.

the fitness function value of 15 particles and compare the value with the local best value. After all the particles find local best value, we choose the global best value among local best values. Then, all the particles move to the directions of weighted combination of local and global best value. We repeated this process up to 30 iterations.

Metric: We used **accuracy** and **fitness** function value to measure the performance. The accuracy is defined as the percentage of correctly clustered documents as in

$$AC_i = \frac{CorrectlyPredict_i}{Predict_i} \tag{8}$$

where $Predict_i$ is the number of documents which are classified as cluster C_i and $CorrectlyPredict_i$ is the number of documents which are belongs to manually generated solution set cluster C_i. Then, the fitness function value is a normalized ratio between the compactness and betweenness as in Equation 7.

Table 2 shows the performance results for *Role Based Particle Swarm Optimization (RoleBasedPSO)* algorithms in terms of accuracy and fitness function value, $f(x)$ at the end of 30 iterations. RoleBasedPSO algorithm shows slightly better performance and approaches faster to global fitness value than traditional PSO algorithm. Figure 5 shows the average accuracy for three different name data set. RoleBasedPSO shows *2.5%* improvement for $Bekkerman$, *3%* for $ECDL$ and *5%* for $Wikipedia$ name data set in terms of accuracy. Considering that $Bekkerman$ name data set is more regularized than other data set, the small improvement is not surprising. RoleBasedPSO performs better when the documents data are more complicated to cluster. Bottom graph shows the accumulated fitness function value for three different name data sets after 30 iterations. The graph shows that RoleBasedPSO reaches to a global minimum faster than traditional PSO by *5%* in terms of normalized fitness value.

4 Related Works

Cui et. el. showed the effectiveness of particle swarm optimization algorithm in document clustering in their paper [16]. Furthermore, the authors showed some improvement by combining the particle swarm optimization with k-means algorithm in [17]. Ghali et. el. proposed an algorithm to change the inertia weight by linear algorithm [18] and by exponential algorithm [19]. The exponential algorithm showed a lower error and failure rate than the linearly reducing particle swarm optimization algorithm but it shows slow convergence. Shi et. el. suggested an optimal parameter selections in particle swarm optimization in [13, 14] which was used in our experiments.

	TraditionalPSO		RoleBasedPSO	
	AC	**f(x)**	**AC**	**f(x)**
Adam Cheyer	.99	2.75	.99	2.74
William Cohen	.91	1.39	.93	1.59
Steve Hardt	.80	2.66	.90	1.74
David Israel	.70	1.87	.76	1.73
Leslie Pack Kaebling	.99	1.55	.99	1.56
Bill Mark	.84	1.96	.88	1.81
Andrew McCallum	.89	1.34	.87	1.48
Tom Mitchell	.83	1.42	.80	1.37
David Mulford	.81	1.63	.80	1.69
Andrew Ng	.81	1.26	.84	1.23
Fernando Pereira	.78	1.55	.82	1.45
Lynn Voss	.85	1.37	.88	1.38
Allan Hanbury	1.0	3.08	1.0	2.92
Andrew Powell	.82	1.36	.79	1.41
Anita Coleman	.95	1.24	.95	1.21
Christine Borgman	.98	1.55	.98	1.32
Donna Harman	.74	2.50	.88	2.64
Edward Fox	.74	2.05	.80	1.43
Gregory Crane	.76	1.41	.76	1.32
Jane Hunter	.93	1.16	.96	1.18
Paul Clough	.77	1.59	.86	1.52
John Kennedy	.78	1.33	.82	1.27
George Clinton	.71	1.47	.76	1.42
Paul Collins	.76	1.29	.81	1.33
Michael Howard	.81	1.38	.80	1.38
Tony Abbott	.66	3.31	.77	2.76
David Lodge	.74	1.79	.77	1.83
Alexander Macomb	.77	1.49	.84	1.52

Table 2. Experimental Performance results. RoleBasedPSO shows better performance and reaches to global minimum faster than PSO. AC represents accuracy and f(x) shows fitness function value at the end of iterations.

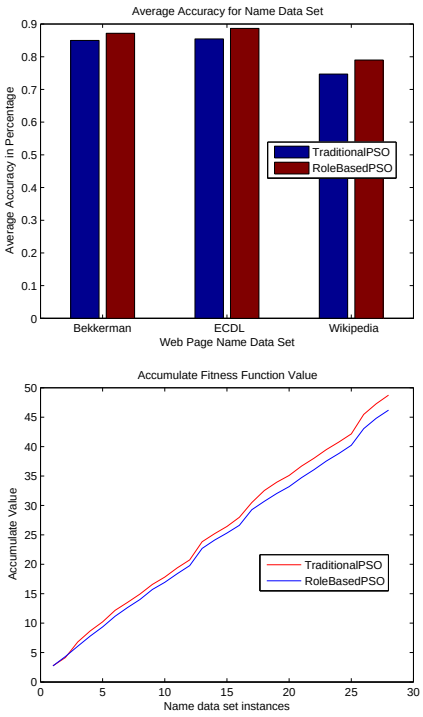

Figure 5. Performance Summary: Top graph shows the average accuracy for three different name data sets; RoleBasedPSO shows *2.5%* **improvement for** *Bekkerman*, *3%* **for** *ECDL* **and** *5%* **for** *Wikipedia* **name data set. Bottom graph shows the accumulated fitness function value for all** *28* **name data sets; RoleBasedPSO algorithm shows** *5%* **improvement.**

To distinguish web appearances of people in a social network, Bekkerman et. al. proposed two algorithms [20]. One is based on link structure of web pages and another algorithm is using multi-way distributional clustering method. Their algorithms show improvement in terms of Fmeasure *Fmeasure is defined as the product of precision and recall*. Minkov et. al. used lazy graph walk algorithm to disambiguate names in email documents in [21]. They provided a framework for email data, where content, social networks and a timeline to integrated in a structured graph. Banerjee et. al. proposed multi-way clustering on relation graphs in [22]. Different types of entities are simultaneously clustered based not only on their intrinsic attribute values, but also on multiple relations between entities. On and Lee used multi-level graph partitioning methods to provide a scalable name disambiguation solution in their paper [23].

In authors awareness, this is the first paper to provide a role based particle swarm optimization algorithm in solving a name data web document clustering problem. In addition, we analyzed the web document characteristics for name entities which are extremely skewed in the aspects of the number of clusters and size of a cluster. Our experiment results show some promising results by assigning different roles to each particle.

5 Concluding Remarks

In this paper, we proposed a role based particle swarm optimization algorithm (RoleBasedPSO) which divides the particles into different groups and assigns different roles to each group. One group is responsible to search a global space, another group is searching for a local space, and the last group stabilizes convergence. Based on our experimental results with three different name data sets (i.e. *Bekkerman, ECDL, Wikipedia*, role based particle swarm

optimization (RoleBasedPSO) algorithm shows *5%* better performance in terms of accuracy with *Wikipedia* name data set. In addition, it requires less number of iterations to reach a global minimum.

In the current RoleBasedPSO, we are using a supervised learning algorithm. However, in the real applications, unsupervised learning algorithm is more suitable. We are planning to modify the current role based particle swarm optimization algorithm to an unsupervised algorithm by combining similarity propagation [24] which is known as a unsupervised clustering algorithm.

References

[1] Zhao and Karypis, "Empirical and theoretical comparisons of selected criterion functions for document clustering," *Machine Learning*, vol. 55(3), pp. pp311–331, 2004.

[2] Al-Sultan and Khan, "Computational experience on four algorithms for the hard clustering problem," *Pattern Recognition, Letter*, vol. 17 (3), pp. 295–308, 1996.

[3] Anderberg, *Cluster Analysis for Applications*. Academic Press, 1973.

[4] Jain, Murty, and Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, 1999.

[5] M. Clerc and J. Kennedy, "The particle swarm-explosion stability and convergence in a multidimensional complext space," *IEEE Transaction on Evolutionary Computation*, vol. 6, pp. 58–73, 2002.

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International joint conference of Neural Networks*, 1995.

[7] ——, *Swarm Intelligence*. Morgan Kaufmann Academic Press, 2001.

[8] Sousa, Silva, and Neves, "Particle swarm based data mining algorithms for classification tasks," *Parallel Computing*, vol. 30, pp. 767–783, 2004.

[9] R. Bekkerman, "Name data set," http://www.cs.umass.edu/ ronb.

[10] "Weps: Searching information about entities in the web," http://nlp.uned.es/weps.

[11] Hartigan, *Clustering Algorithms*. John Wiley and Sons, 1975.

[12] D. Zeimpekis and E. Gallopoulos, *TMG: A MATLAB toolbox for generating term document matrices from text collections*, 2006.

[13] R. Eberhart and Y. Shi, "Comparing inertia weights and constricting factors in particle swarm optimization," *Congress on Evolutionary Computing*, vol. 1, pp. 84–88, 2000.

[14] Y. Shi and R. Eberhart, "Parameter selection in paticle swarm optimization," in *The 7th Annual Conference on Evolutionary Programming, San Diego, CA*, 1998.

[15] Li-ping, Huan-jun, and Shang-xu, "Optimal choice of parameters for particle swarm optimization," *Journal of Zhejiang University Science*, vol. 6(A), pp. 528–534, 2004.

[16] X. Cui, T. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," in *Proceedings of Swarm Intelligence Symposium 2005*, 2005.

[17] X. Cui. and T. Potok, "Document clustering analysis based on hybrid pso+kmeans algorithm," *Journal of Computer Sciences (special issue)*, pp. 27–33, 2005.

[18] N. El-Dessouki, N. Ghali, and M. Zaki, "A new approach to weight variation in swarm optimization," in *Proceedings of Al-azhar Engineering, the 9th International Conference*, 2007.

[19] N. Ghali, N. El-Dessouki, M. Zaki, and L. Bakrawi, "Exponential particle swarm optimization approach for improving data clustering," in *Proceedings of World Academy of Science, Engineering and Technology*, 2008.

[20] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *Proceedings of International World Wide Web Conference Committee*, 2005.

[21] E. Monkov, W. Cohen, and A. Y. Ng., "Contextual search and name disambiguation in email using graphs," in *Proceedings of SIGIR*, 2006.

[22] A. Banerjee, S. Basu, and S. Merugu, "Multi-way clustering on relation graphs," in *Proceedings of SIAM Data Mining 2007*, 2007.

[23] B. On and D. Lee, "Scalable name disambiguation using multi-level graph partition," in *Proceedings of SIAM Data Minings*, 2006.

[24] Frey and Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, 2007.