2008 IEEE Swarm Intelligence Symposium
St. Louis MO USA, September 21-23, 2008

Dimensionality Reduction Particle Swarm Algorithm for High Dimensional Clustering

Xiaohui Cui, Justin M. Beaver, Jesse St. Charles and Thomas E. Potok

Abstract—**The Particle Swarm Optimization (PSO) clustering algorithm can generate more compact clustering results than the traditional K-means clustering algorithm. However, when clustering high dimensional datasets, the PSO clustering algorithm is notoriously slow because its computation cost increases exponentially with the size of the dataset dimension. Dimensionality reduction techniques offer solutions that both significantly improve the computation time, and yield reasonably accurate clustering results in high dimensional data analysis. In this paper, we introduce research that combines different dimensionality reduction techniques with the PSO clustering algorithm in order to reduce the complexity of high dimensional datasets and speed up the PSO clustering process. We report significant improvements in total runtime. Moreover, the clustering accuracy of the dimensionality reduction PSO clustering algorithm is comparable to the one that uses full dimension space.**

I. Introduction

The clustering of high dimension data sets is a process that is needed in many application areas. Because traditional data clustering algorithms tend to be biased towards local optimum when applied to high dimension data sets, Particle Swarm Optimization (PSO) has been used for solving data clustering problems in recent years [1-5]. Many researchers [1, 5, 6] have indicated that when utilizing the PSO algorithm's optimal ability, and given sufficient time, PSO generates a more compact clustering result from low dimensional data than the traditional K-means clustering algorithm. However, when clustering high dimensional datasets, the PSO clustering algorithm is notoriously slow because the algorithm needs to repeatedly compute high dimension data vector similarities.

Researchers have found that dimensionality reduction techniques offer solutions that both improve the computation time, and still yield accurate results in some high dimensional data analysis [7-11]. It is therefore highly desirable to reduce the dimensionality of a high dimension dataset before clustering in order to maintain tractability. Since reducing dataset dimensionality may result in a loss of information, the lower dimension representation generated by a dimensionality reduction algorithm must be a good approximation of the full high dimensional dataset.

In this paper, we introduce our research of applying two different dimensionality reduction techniques as data pre-processing for the PSO clustering algorithm in order to reduce the complexity of the high dimensional dataset and speed up the clustering computation time. We use the document dataset as our experimental data. The document Vector Space Model (VSM), originated by G. Salton [12], represents documents as vectors in a vector space. Compared to other high dimension datasets, document datasets' dimensionality value is usually quite large, on the order of 10,000 dimensions for moderately-sized document collections.

The document vector set comprises an $\boldsymbol{n \times m}$ term-document matrix $\boldsymbol{X}$, in which each row of the matrix represents a document, and each entry $\boldsymbol{x(i,j)}$ represents the weighted frequency of term $\boldsymbol{i}$ in document $\boldsymbol{j}$. The entry weight value represents the significance of this term in a document. To calculate the term weight, the frequency of the term within a document and in the entire set of documents must be considered. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF) [13]. The weight of term $\boldsymbol{i}$ in document $\boldsymbol{j}$ is given in Eq. (1):

$$x_{ji} = tf_{ji} * idf_{ji} = tf_{ji} * \log_2(n / df_{ji}) \tag{1}$$

where tf_{ji} is the number of occurrences of term i in the document j; df_{ji} indicates the term frequency in the collections of documents; and n is the total number of documents in the collection. This weighting scheme discounts the frequent words with little discriminating power. A major benefit of this VSM document representation is that methods developed in linear algebra for use in other application areas can be applied to document retrieval as well. However, each unique term in the document collection represent one dimensionality of the vector space. Modern document sets sometimes includes millions of documents. To achieve higher efficiency in clustering the VSM dataset, it is often necessary to reduce the dimension severely.

Traditional document clustering involves dividing a set of documents into a specified number of clusters [14]. The motivation behind clustering a set of data is to find inherent structure in the data and expose this structure as a set of groups. The data objects within each group should exhibit a large degree of similarity while the similarity among

Manuscript received May 28, 2008; revised July 19, 2008. This work was supported in part by in part by the office of Naval Research (N0001408IP20066).

Xiaohui Cui is with the Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA (phone: 865-576-9654; fax: 865-241-0003; e-mail: cuix@ornl.gov).

Jesse St. Charles was with University of Tennessee in Chattooga, Chattooga, TN, USA. He is now with the Center for Computational Analysis of Social and Organizational Systems, Carnegie Mellon University, Pittsburgh Pennsylvania, USA.

Justin M. Beaver and Thomas E. Potok are with Computational Sciences and Engineering Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

978-1-4244-2705-5/08/$25.00 ©2008 IEEE

different clusters should be minimized [15]. A lower dimension approximation of document vector can significantly increase the clustering speed. The two most popular dimensionality reduction methods, Latent Semantic Indexing (LSI) and Random Projection (RP), are investigated in this research for reducing the dimensionality of the document vector space. This should in theory speed up similarity computation times in PSO clustering.

In the following sections, an introduction to the PSO clustering algorithm is given. Dimensionality reduction and the LSI and RP techniques are discussed. The experimental design of the hybrid PSO (HPSO) clustering algorithm is described. The experiment results of applying different dimensionality reduction techniques as PSO algorithm's data pre-processing are compared. Finally, we present concluding remarks and analytical derivations for the proposed technique.

II. PARTICLE SWARM CLUSTERING ALGORITHM

In clustering research, it is common to view the clustering problem as an optimization problem that locates optimal centroids rather than optimal partitions. This view offers us a chance to apply the PSO algorithm to the clustering solution. The objective of the PSO clustering algorithm is to discover centroids of clusters for minimizing the intra-cluster distance as well as maximizing the distance between clusters. Omran [16] used the PSO clustering algorithm for clustering image data. In [1], a K-means+PSO clustering algorithm was proposed and evaluated on the standard UCI clustering test datasets. This k-menas+PSO clustering algorithm uses k-means to pre-process and initialize the PSO positions.

In [6], a PSO clustering algorithm for documents is proposed. In this PSO document clustering algorithm, the multi-dimensional document vector space is modeled as a problem space. Each term in the document dataset represents one dimension of the problem space. Each document vector can be represented as a dot in the problem space. One particle's position in the swarm represents one possible solution for clustering the document collection. Therefore, a swarm represents a number of candidate clustering solutions for the document collection. Each particle maintains a matrix $X_i = (C_1, C_2, \ldots, C_i, .., C_k)$, where C_i represents the ***ith*** cluster centroid vector and k is the number of clusters.

$$v_{id} = w * (v_{id} + c_1 * rand_1 * (p_{id} - x_{id}) + c_2 * rand_2 * (p_{gd} - x_{id})) \quad (2a)$$

$$x_{id} = x_{id} + v_{id} \quad (2b)$$

The PSO clustering algorithm can be represented in eq (2a) and (2b). As indicated in Eq(2a), each particle acquires experience by moving through the solution space with velocity v_{id}. According to this experience and those of its neighbors, the particle adjusts the centroid vector's position in the vector space at each generation. The two random values (***rand1, rand2***) are generated each generation. These values along with the inertial weight factor w provide the necessary diversity to the swarm, avoiding stagnation at local optima. Also, the greater the number of particles used, the quicker the system converges to a global optimum. The behavior of the PSO clustering algorithm can be classified into two stages: a global searching stage and a local refining stage. Initial PSO search iterations can be classified as a global searching stage. After several iterations, the particle's velocity will gradually reduce and the particle's exploration area will shrink as the particle approaches the optimal solution. The global searching stage gradually merges into the local refining stage. By selecting different parameters in the PSO algorithm, we can control the shift time from the global searching stage to the local refining stage. The later this shift occurs, the higher the possibility that it can find a globally optimal solution. Although the PSO algorithm generates much better clustering result than the K-means algorithm does, it is much slower to execute because of the high dimensionality of the document dataset. In [3], a hybrid PSO (HPSO) that uses PSO for global search and then employs K-means for local refining is presented. This hybrid PSO significantly increases the result accuracy and reduces the runtime.

The PSO clustering algorithm needs a fitness function to evaluate each particle's performance at each iteration. In most data clustering PSO algorithms, a common approach is to use a cluster validity measure index functions as a fitness function. In [1], the sum of squared error (SSE) index is used as the fitness function. In [3], a modified SSE index, the average distance of document to cluster centroid (ADDC), is used to evaluate the particle's performance in document clustering. There are many other clustering evaluation criteria that have been developed. The common goal of the clustering evaluation is to identify compact clusters that are well separated. This goal can be represented as two criteria:

Compactness: The similarity distance between the elements in one cluster. It indicates the compact degree of the cluster. The elements in one cluster should be similar to each other.

Separation: The similarity distance between centroids of different cluster. The clusters should be well separated from each other.

These two criteria have been widely used to measure the quality of clustering result. The PSO clustering fitness function in this research is the combination of these two criteria.

III. DIMENSIONALITY REDUCTION

Dimensionality reduction is an important task in data mining for the classification, clustering, and visualization of high dimensional data by mitigating undesired properties of high dimensional space. More simply, it retains the most important attributes in a dataset and removes the noise. In recent years, a large number of new techniques for dimensionality reduction have been proposed. A systematic review of different dimensionality reduction techniques has been presented in [17].

Dimensionality reduction techniques can be classified into two categories: feature extraction and feature transformation. For textual data, the feature extraction technique involves specifying the bounds on the term distribution in the collection, i.e., to consider only terms which occur between the predefined minimum and maximum number of documents in the collection. The rationale is that terms that are present in a large percentage of documents in the collection are uninformative while those that are extremely rare can be disposed off as noise. Such words are therefore ignored. The feature extraction methods are usually tailored according to the nature of the data, and therefore are not generally applicable in all data mining or process tasks. The feature transformation techniques perform a transformation of the high dimensional vector data into a meaningful representation in lower dimensional subspace where the new dimension can be viewed as a linear transform from the original dimensions. Suppose we have an $\boldsymbol{n \times m}$ matrix $\boldsymbol{X}$ with $\boldsymbol{n}$ data vectors $\boldsymbol{x_i}$ and dimensionality $\boldsymbol{m}$, a feature transformation technique transforms matrix $\boldsymbol{X}$ into a new $\boldsymbol{n \times d}$ matrix $\boldsymbol{Y}$ with dimensionality $\boldsymbol{d}$ ($\boldsymbol{d<m}$), while retaining the major feature of matrix $\boldsymbol{X}$ as much as possible. In this section, the mathematic details of two different dimensionality reduction techniques, Latent Semantic Indexing and Random Projection are introduced.

A. Latent Semantic Indexing

Latent Semantic Indexing (LSI) [18] (or Latent Semantic Analyzing (LSA)), is one of the standard dimension reduction techniques in information retrieval. It analyzes relationships between documents and document terms in the matrix by producing a set of concepts related to the document and terms. LSI uses Singular Value Decomposition (SVD) [19] to embed the original high dimensional space into a low dimensional space with minimal distance distortion. In SVD, an $\boldsymbol{n \times m}$ matrix $\boldsymbol{X}$ can be decomposed into three matrixes of special forms:

$$X_{[n\times m]} = U_{[n\times n]} \cdot S_{[n\times r]} \cdot V^T_{[m\times m]} \tag{3}$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrixes that contain the left and right singular vectors of $\boldsymbol{X}$**.** They are often referred as the term projection matrix and the document projection matrix, respectively. $\boldsymbol{S}$ is the diagonal matrix that contains the singular values of $\boldsymbol{X}$, and the subscript $\boldsymbol{r}$ denotes the number of singular values. The singular values are sorted in descending order to indicate their "importance" or rank in the matrix. The dimensionality of the data can be reduced by projecting the data onto the space spanned by the left singular vectors corresponding to the k largest singular values. The top rank-$\boldsymbol{k}$ approximation of $\boldsymbol{X}$ can be generated by:

$$X_{[n\times k]} = U^T_{[n\times k]} \cdot X_{[n\times m]} \tag{4}$$

In this way, $\boldsymbol{X_{[n\times m]}}$ is projected from $\boldsymbol{m}$ dimensional space to $\boldsymbol{k}$ dimensional space $\boldsymbol{X_{[n\times k]}}$ (normally $\boldsymbol{m>>k}$) and the top $\boldsymbol{k}$ "importance" features are retained. LSI not only can retain the most important relationship between terms and documents, it is also effective at removing noise and redundancy from the dataset [20].

A major drawback of LSI is its high computational cost. For a data matrix with $\boldsymbol{n \times m}$ elements, the time complexity of LSI dimensionality reduction is $\boldsymbol{O(m^2n)}$. However, for a sparse matrix, which is commonly used in representing document term frequencies, the computational complexity can be reduced to $\boldsymbol{O(cmn)}$, where $\boldsymbol{c}$ is the average number of non-zero elements in each vector. Regardless, it is computational expensive when LSI is applied on a large dataset.

B. Random Projection

Random projection (RP) [10] is another feature transformation-based dimensionality reduction technique. It is much less computationally expensive than LSI. The idea of RP is motivated by Johnson and Lindenstrauss Lemma [11] which states that a set of $\boldsymbol{n}$ points in high dimensional space can be mapped onto an $\boldsymbol{O(log(n)/e^2)}$ space and that the distance between the points is still approximately preserved with distortion of no more than a factor of $\boldsymbol{(1+-e)}$, for any $\boldsymbol{0<e<1}$. In RP, resulting low dimension datasets are generated randomly with no importance placed on order. As shown in Equation 3, the high dimensional data space $\boldsymbol{X_{[n\times m]}}$ can be projected to low dimensional space $\boldsymbol{A_{[n\times k]}}$ by using a randomly generated projection matrix $\boldsymbol{R}$ of size $\boldsymbol{k \times m}$.

$$A_{[n\times k]} = X_{[n\times m]} \cdot R_{[m\times k]} \tag{5}$$

There are many ways to build this random matrix $\boldsymbol{R}$. Typically, each element of $\boldsymbol{R}$ can be Gaussian distributed, $\boldsymbol{R_{ij}=N(0,1)}$. But the least expensive way to build this matrix is:

$$R_{ij} = \sqrt{3} * \begin{cases} -1 & \mid probability = 1/6 \\ 0 & \mid probability = 2/3 \\ -1 & \mid probability = 1/6 \end{cases} \tag{6}$$

Since this process involves only one matrix multiplication step, for a sparse matrix $\boldsymbol{X}$, the computational complexity of RP is $\boldsymbol{O(cnk)}$, where $\boldsymbol{c}$ is the average term number of each document and $\boldsymbol{k}$ is the reduced dimension number. RP can maintain the distance information in a high dimensional dataset, which makes it useful in situations where the distance between data points is meaningful, such as data clustering. Intuitively, it may seem surprising that random projection can reduce the dimensionality of the data in a manner that preserves the approximate structure of the original dataset. The mathematic proof of the RP dimensionality reduction technique has been discussed in [18].

IV. EXPERIMENTS AND RESULTS

We investigated the relative effectiveness of LSI and RP when used for data pre-preprocessing of the HPSO clustering algorithm on a benchmark dataset. These two dimensionality reduction based HPSO clustering algorithms are named LSI-HPSO and RP-HPSO, respectively in our experiments. We run each dimensionality reduction technique to reduce the dataset to different dimension $\boldsymbol{k}$ values. We apply clustering on the data before and after the dimensionality reduction to verify and compare the results. The accuracy of the clustering results and the runtime of the algorithms will be compared. The runtime for the dimensionality reduction based HPSO algorithm will include the dimensionality reduction time and the HPSO clustering time.

A. Experimental Setup

The HPSO clustering algorithm [3] that combines the PSO clustering algorithm with K-means clustering algorithm is implemented for the experiments. The HPSO algorithm includes two modules, a PSO module and a K-means module, which performs the global search stage and localized refinement stage, respectively. In the initial stage, the PSO module is executed for a short period (50 iterations) to discover the vicinity of the optimal solution by a global search, and to minimize resource consumption. The result from the PSO module is then used as the initial seed of the K-means module. The K-means algorithm will then be applied for refining and generating the final result. In the PSO module, we choose 20 particles with an inertia weight $\boldsymbol{w}$ initially set as 0.95 and the acceleration coefficient constants $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are set to 1.4. These values are chosen based on [3].

B. Datasets

We used the document dataset from the CLUTO toolkit [21] to compare the performance of different dimensionality reduction techniques when these techniques are combined with the HPSO algorithms. These document datasets are derived from the Text REtrieval Conference (TREC) collections. In these document datasets, very common words (e.g. function words: “a”, “the”, “in”, “to”; pronouns: “I”, “he”, “she”, “it”) are stripped out completely and different forms of a word are reduced to one canonical form by using Porter’s stemming algorithm [22]. In order to reduce the impact of the length variations of different documents, each document vector is normalized so that it is of unit length. The document collection was converted into a vector space matrix of dimension 414×6429. The dimensionalities of the matrix are then reduced to lower $\boldsymbol{k}$-dimensions. The cosine similarity was used as the vector distance metric.

C. Clustering Results Evaluation Methods

The experiment results should be evaluated using an informative quality measurement to reflect the quality of the clustering results. Since the document collection datasets used in this experiment have already been classified by human experts, we will use the human classification results as a standard to evaluate these three HPSO clustering algorithms. We use the F-measure as the quality measure. This F-measure combines the precision and the recall ideas from information retrieval literature. The precision P and recall R of a cluster j (generated by the clustering algorithm) with respect to a class i (prior knowledge of the datasets) is defined as:

$$P(i,j) = \frac{N_{ij}}{N_i} \tag{7}$$

$$R(i,j) = \frac{N_{ij}}{N_j} \tag{8}$$

Where N_{ij} is the element number of class i within cluster j, N_j is the number of items of cluster j and N_i is the number of members of class i. The corresponding value of the F-measure is:

$$F(i) = \frac{2PR}{P+R} \tag{9}$$

With respect to class *i*, members of *i* may be organized into different clusters. That will generate multiple F-measure value for class *i*. We consider the cluster with the highest F-measure score as the cluster for class *i*. The overall F-measure for the clustering result of one algorithm is computed as:

$$F = \frac{\sum_{i=1}^{n} |i| * F(i)}{\sum_{i}^{n} |i|} \tag{10}$$

Where n is the number of the clusters in the dataset and $|i|$ is the number of data objects in class *i*. The F value is limited within the interval [0, 1] with the higher the F-measure producing the better the clustering result.

D. Experimental Results

All experiment codes are implemented in MATLAB. The experiments are conducted on a Pentium 4, 2GB memory desktop. We conducted a set of experiments with 20 different choices of dimensionality k = {10, 20, 30, 40, 50, 100, 150 …. 800}, and compared their results. The same experiments are repeated for each algorithm: LSI-HPSO, RP-HPSO and the original HPSO. The CPU time and the clustering results of each experiment are collected. Each clustering algorithm is repeated 20 times and the average result at each *k* were chosen as the final result. Figure 1 summarizes the clustering accuracy for the three algorithms on different dimensionality $\boldsymbol{k}$. The top level line is the F measure value of the HPSO clustering the original document matrix dataset. The result line is expanded to serve as the reference. The other two curves show the F-measure of results obtained by LSI-HPSO and RP-HPSO. Based on the results, we observed that LSI achieves superior results with low dimensionalities. Within the dimension range of [10, 150], LSI-HPSO clustering accuracy performance is even better than HPSO clustering on original data. The RP-HPSO has low accuracy performance in low dimension range of [20, 400]. It can achieve related better accuracy performance when reduced data has more than 400 dimensionalities. But its result f measure is still lower than the clustering result on original data.

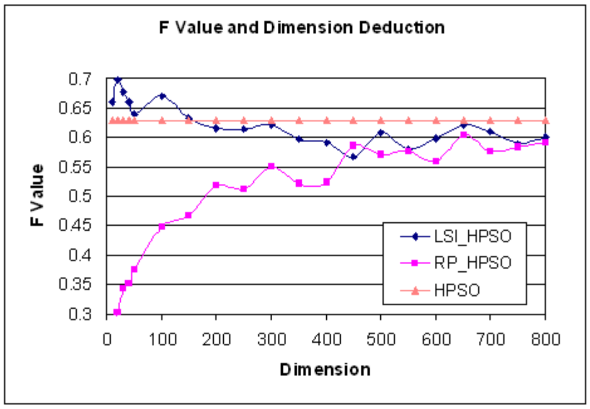

Fig. 1: The accuracy performance of HPSO clustering algorithm with different dimensionality reduction techniques

Figure 2 displays the CPU runtime for HPSO clustering on different kind of data. The top level line is the CPU running time for HPSO clustering on original data. It has highest running time value, around 140 seconds for clustering 414 documents. The middle curve is the CPU time for LSI-HPSO clustering on different dimensionalities reduced data. Runtimes for the different k values remain relatively stable because the major computational cost is the LSI dimensionality reduction part. Because the RP dimensionality reduction is much faster than the LSI, RP-HPSO uses the fewest CPU time compare to other two algorithms. The CPU running time for different dimensionalities reduced data ranges from below one second for 20 dimensionality reduced data to around 10 seconds for 800 dimensionality reduced data.

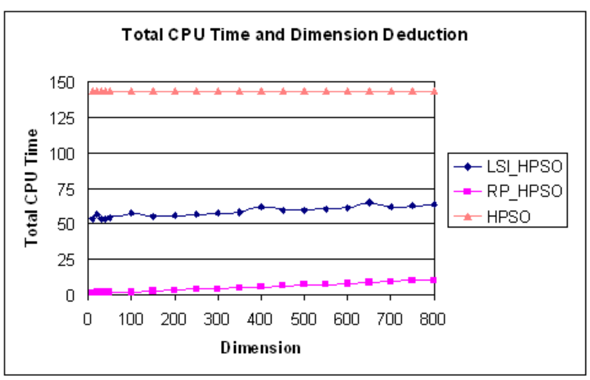

Fig. 2: The CPU running time of HPSO clustering algorithm with different dimensionality reduction techniques

V. Discussion

From our results it is clear that LSI-HPSO can generate better F-measure results in low dimension data than HPSO clustering on original data. It has also been shown that the LSI dimensionality reduction technique not only reduces the dimensions, but also effectively removes noise and redundancy in the dataset. It enhances the hidden semantic structure of the document dataset. Previous research has indicated that the choice of ***k*** is important to the performance of LSI dimensionality reduction [18]. Our HPSO document clustering experiments has revealed that the LSI can preserve the document similarities sufficiently for very small ***k*** values as compared to the original dimensionality ***m***. LSI-HPSO reaches its highest F-measure value when *k* equals 20. In our experiments, the LSI-HPSO algorithm performs nearly three times faster than HPSO on original data. But RP-HPSO can run even faster. It is 26 times faster than HPSO and 10 times faster than LSI-HPSO at 400 dimensionalities. If accuracy is important, LSI-HPSO clustering on twenty dimension data may generate a good result. If clustering time is critical, the RP-HPSO with higher dimension data will be suitable.

Rapadimitriou [23] proposed to combine RP and LSI using RP to initially reduce a high dimensional dataset to an intermediate dimensionality on which LSI could then perform an additional dimensionality reduction process. In theory, this method might reduce the computational cost but, as indicated by Lin [24], the computational cost will increase for document matrix dimensionality reduction. Because the document matrix is represented as a sparse matrix, the matrix produced by RP will no longer be sparse and the LSI computational cost will thus increase significantly.

VI. CONCLUSIONS

In this research, we investigated the performance of HPSO clustering algorithms integrated with different dimensionality reduction techniques. We report significant improvements in total clustering runtime for two dimensionality reduction techniques. Moreover, the clustering accuracy of the LSI-HPSO and RP_HPSO clustering algorithm is comparable to the one that uses the full term space. In some cases, LSI can even help improve the accuracy of HPSO clustering algorithm. The major drawback of LSI-HPSO is that the LSI dimensionality reduction process is computational expensive compared to the RP dimensionality reduction process.

The experiments presented in this paper focus on improving the performance of HPSO document clustering by transforming the high dimensional vector document matrix into a lower dimension matrix. The dimensionality reduction idea can also be used to improve the HPSO clustering algorithm for other kinds of high dimensional data. According to our experiments, when LSI combines with the HPSO clustering algorithm, the document clustering result is more accurate and the total runtime is three times faster than HPSO clustering without the dimensionality reduction. Combining the RP technique with the HPSO clustering algorithm can be 26 times faster than just the HPSO and still keep reasonable accurate result. Future research includes implementing more dimensionality reduction techniques and applying them on different high dimensional datasets to discover the most efficient technique.

ACKNOWLEDGMENTS

Oak Ridge National Laboratory is managed by UT-Battelle LLC for the US Department of Energy under contract number DE-AC05_00OR22725. This work was supported in part by the office of Naval Research (N0001408IP20066). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Office of Naval Research, the Department of Energy or the U.S. government.

REFERENCES

[1] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *2003 Congress on Evolutionary Computation, 8-12 Dec. 2003*, Canberra, ACT, Australia, 2003, pp. 215-220.

[2] X. Zhang, J. Wang, Z. Fan, and B. Li, "Spatial clustering with obstacles constraints using ant colony and particle swarm optimization," Nanjing, China, 2007, pp. 344-356.

[3] X. Cui and T. E. Potok, "Document Clustering Analysis Based on Hybrid PSO+K-means Algorithm," *Journal of Computer Sciences,* vol. Special Issue, pp. 27-33, 2005.

[4] W. Tong, L. Da-Xin, L. Xuan-Zuo, S. Wei, and A. Gufran, "Clustering large scale of XML documents," Taichung, Taiwan, 2006, pp. 447-55.

[5] M. G. H. Omran, A. Salman, and A. P. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Analysis and Applications,* vol. 8, pp. 332-44, 2006.

[6] X. Cui, T. E. Potok, and P. Palathingal, "Document clustering using particle swarm optimization," in *2005 IEEE Swarm Intelligence Symposium, 8-10 June 2005*, Pasadena, CA, USA, 2005, pp. 185-191.

[7] C. Ding, H. Xiaofeng, Z. Hongyuan, and H. D. Simon, "Adaptive dimension reduction for clustering high dimensional data," Maebashi City, Japan, 2002, pp. 147-54.

[8] K. V. Ravi Kanth, D. Agrawal, and A. Singh, "Dimensionality reduction for similarity searching in dynamic databases," Seattle, WA, USA, 1998, pp. 166-76.

[9] V. Vinay, I. J. Cox, K. Wood, and N. Milic-Frayling, "A comparison of dimensionality reduction techniques for text retrieval," Los Angeles, CA, USA, 2005, p. 6 pp.

[10] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," San Francisco, CA, USA, 2001, pp. 245-50.

[11] S. Kaski, "Dimensionality reduction by random mapping: Fast similarity computation for clustering," Anchorage, AK, USA, 1998, pp. 413-418.

[12] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," Cornell Univ., Ithaca, NY, USA, Copyright 1975, IEE CU-CSD-74-218, 1974.

[13] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing & Management,* vol. 24, pp. 513-523, 1988.

[14] M. R. Anderberg, *Cluster Analysis for Applications*. New York: Academic Press, Inc., 1973.

[15] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys,* vol. 31, pp. 264-323, 1999.

[16] M. G. H. Omran and A. P. Engelbrecht, " Image classification using particle swarm optimization," in *4th Asia Pacific Conference on Simulated Evolution and Learning*, Singapore, 2002, pp. 370-374.

[17] I. K. Fodor, "Survey of Dimension Reduction Techniques," United States 2002.

[18] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society for Information Science,* vol. 41, p. 391, 1990.

[19] M. T. Heath, "Sparse matrix computations," Las Vegas, NV, USA, 1984, pp. 662-5.

[20] Z. Xiaoyan, M. W. Berry, and P. Raghavan, "Level search schemes for information filtering and retrieval," *Information Processing & Management,* vol. 37, pp. 313-34, 2001.

[21] Y. Zhao and G. Karypis, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery,* vol. 10, pp. 141-68, 2005.

[22] M. F. Porter, "An algorithm for suffix stripping," *Program,* vol. 14, pp. 130-137, 1980.

[23] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, "Latent semantic indexing: a probabilistic analysis," Seattle, WA, USA, 1998, pp. 159-68.

[24] J. Lin and D. Gunopulos, "Dimensionality reduction by random projection and latent semantic indexing," in *3rd SIAM International Conference on Data Mining*, 2003.