International Journal of the Physical Sciences Vol. 7(1), pp. 116 - 120, 2 January, 2012
Available online at http://www.academicjournals.org/IJPS
DOI: 10.5897/IJPS11.692
ISSN 1992-1950 ©2012 Academic Journals

Full Length Research Paper

Text clustering on latent semantic indexing with particle swarm optimization (PSO) algorithm

Eisa Hasanzadeh[1], Morteza Poyan rad[2]* and Hamid Alinejad Rokny[3]

[1]Electrical and Computer Engineering Faculty, Qazvin Islamic Azad University, Qazvin, Iran.
[2]Electrical and Computer Engineering Faculty, Qazvin Islamic Azad University, Member of Young Research Club, Qazvin, Iran.
[3]Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

Accepted 16 November, 2011

Most of web users use various search engines to get specific information. A key factor in the success of web search engines are their ability to rapidly find good quality results to the queries that are based on specific terms. This paper aims at retrieving more relevant documents from a huge corpus based on the required information. We propose a particle swarm optimization algorithm based on latent semantic indexing (PSO+LSI) for text clustering. PSO family of bio-inspired algorithms has recently successfully been applied to a number of real word clustering problems. We use an adaptive inertia weight (AIW) that do proper exploration and exploitation in search space. PSO can merge with LSI to achieve best clustering accuracy and efficiency. This framework provides more relevant documents to the user and reduces the irrelevant documents. It would be seen that for all numbers of dimensions, PSO+LSI are faster than PSO+Kmeans algorithms using vector space model (VSM). It takes 22.3 s for PSO+LSI method with 1000 terms to obtain its best performance on 150 dimensions.

Key words: Vector space model, particle swarm optimization (PSO) algorithm, latent semantic indexing, text clustering, adaptive inertia weight.

INTRODUCTION

Clustering means the act of partitioning an unlabeled dataset into groups of similar objects. Each group, called a ‘cluster’, consists of objects that are similar between themselves and dissimilar to objects of other groups. In the past few decades, cluster analysis has played a central role in a variety of fields ranging from engineering (machine learning, artificial intelligence, pattern recognition, mechanical engineering, electrical engineering), computer sciences (web mining, spatial database analysis, textual document collection, image segmentation), life and medical sciences (genetics, biology, microbiology, paleontology, psychiatry, pathology), to earth sciences (geography. geology, remote sensing), social sciences (sociology, psychology, archeology, education), and economics (marketing, business) (Evangelou et al., 2001; Lillesand and Keifer, 1994; Rao, 1991; Duda and Hart, 1993; Fukunaga, 1990; Cui et al., 2005).

Clustering algorithms can be categorized as either hierarchical or partitioning. Hierarchical clustering techniques proceed by either a series of successive merges or a series of successive divisions. The result is the construction of a tree like structure or hierarchy of clustering’s which can be displayed as a diagram known as a dendogram (Sorg and Cimiano, 2008; Errecalde et al., 2008).

Agglomerative hierarchical methods begin with the each observation in a separate cluster. These clusters are then merged, according to their similarity (the most similar clusters are merged at each stage), until only one cluster remains. Divisive hierarchical methods work in the opposite way. An initial cluster containing all the objects are divided into sub-groups (based on dissimilarity) until

*Corresponding author. E-mail: alinejad.h@umz.ac.ir or alinejad.hamid@gmail.com. Tel: +9809111266458.

each object has its own group. Agglomerative methods are more popular than divisive methods (Sorg and Cimiano, 2008; Pinto et al., 2006).

The K-means and its variants represent a category of partitioning clustering algorithms that create a flat, non-hierarchical clustering that consists of *K* clusters. The K-means algorithm iteratively refines a randomly chosen set of *K* initial centroids, minimizing the average distance (maximizing the similarity) of documents to their closest (most similar) centroid. It is an iterative hill-climbing algorithm and solution suffering from the limitation of the sub optimal which is known to depend on the choice of initial clustering distribution.

In addition to the K-means algorithm, several algorithms, such as genetic algorithm (GA) (Gareth et al., 1995; Raghavan and Birchand, 1999) and self-organizing maps (SOM) (Merkl, 2002), have been used for document clustering. Particle swarm optimization (PSO) (Kennedy et al., 2001; Cagnina et al., 2008; Passaro and Starita, 2008) is another computational intelligence method that has already been applied to Document clustering. A hybrid PSO + kmeans algorithm (Cui et al., 2005) performs a search in complex and large landscape and provide near optimal solutions for objective or fitness function of an optimization problem. However the cost of computational time is high because its long representation evolves in high dimensional space. Hybrid PSO algorithm use vector space model (VSM), it need a large number of features to represent high dimensions and it is not suitable for hybrid PSO since the cost of computational time will be high.

LATENT SEMANTIC INDEXING (LSI)

LSI as one of the standard dimension reduction techniques in information retrieval has enjoyed long lasting attention (Ding, 2000; Berry et al., 1995; Deerwester et al., 1990). By detecting the high-order semantic structure (term-document relationship), it aims to address the ambiguity problem of natural language, the use of synonymous and polysemous words therefore a potentially excellent tool for automatic indexing and retrieval.

LSI uses singular value decomposition (SVD) to embed the original high dimensional space into a lower dimensional space with minimal distance distortion in which the dimensions in this space are orthogonal (statistically uncorrelated). During the SVD process the newly generated dimensions are ordered by their importance. Using the full rank SVD the term-document matrix A is decomposed as $A=USV^T$ where S is diagonal matrix containing singular values of A. U and V are orthogonal matrices containing left and right singular values of A, often referred to as term projection matrix respectively. Using truncated SVD the best rank-k approximation of A is $A_k \approx U_k S_k V_k^T$ in which A is projected from m dimensional space to k dimensional space (m>k). The truncated SVD not only captures the most important associations between terms and documents but also effectively removes noise and redundancy and word ambiguity within the dataset (Deerwester et al., 1990).

The transformed LSI for document representation

Here we use a transform of the original LSI to construct a corpus-based document representation which can appropriately reveal the associative semantic relationship between documents. A document d is initially represented as a m×1 matrix, where m is the number of terms. Because matrix U represents the matrix of terms vectors in all documents and the proper number of rank U_k spans the basis vectors of U. In our approach we use the multiplying of matrices d^T and U_k to represent the document vector. So each document vector is defined by: $d'=d^TU_k$. And the corpus can be newly organized by: $C=DU_k$, where *D* is the document-by-term matrix. The reduced space hopefully captures the true relationships between documents. Our approach of transformed LSI model for corpus representation is shown in Figure 1.

PSO ALGORITHM FOR TEXT CLUSTERING

Swarm intelligence (SI) is an innovative distributed intelligent paradigm for solving optimization problems that originally took its inspiration from the biological examples by swarming, flocking and herding phenomena in vertebrates. Particle Swarm Optimization (PSO) incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the idea is emerged (Kennedy et al., 2001; Cagnina et al., 2008; Passaro and Starita, 2008). PSO is a population-based optimization tool, which could be implemented and applied easily to solve various function optimization problems. As an algorithm, the main strength of PSO is its fast convergence, which compares favorably with many optimization algorithms like GA (Gareth et al., 1995; Song and Park, 2006), simulated annealing (SA) (Sneath and Sokal, 1973) and other optimization algorithms (Ercan, 2008). For applying PSO successfully, one of the key issues is finding how to map the problem solution into the PSO particle, which directly affects its feasibility and performance.

Bird flocking optimizes a certain objective function. Each particle knows its best value so far (pbest) and its position. This information is an analogy of personal experiences of each particle. Moreover, each particle knows the best value so far in the group (gbest) among pbests. This information is analogy of knowledge of how the other particles around them have performed. Namely, each particle tries to modify its position using the following information:

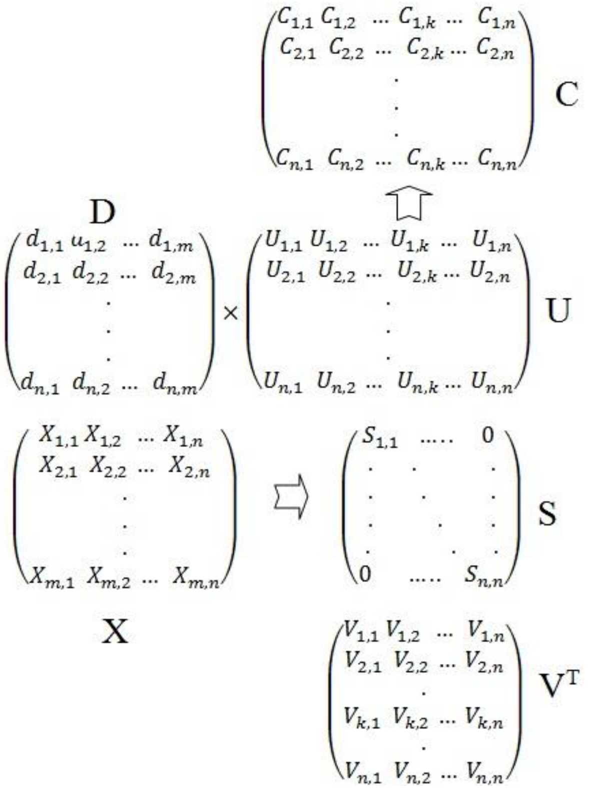

Figure 1. LSI model for document representation.

1. Current positions.
2. Current velocities.
3. Distance between the current position and pbest.
4. Distance between the current position and gbest.

This modification can be represented by the concept of velocity. Velocity position of each particle can be modified by the following equation:

$V_{id}=W*V_{id}+C_1*rand()*(P_{id}-X_{id})+C_2*rand()*(P_{gd}-X_{id})$ (1)

$X_{id}=X_{id}+V_{id}$ (2)

V_{id}, velocity of particle; X_i, current position of particle; *W*, inertia weight; C_1 and C_2, determine the relative influence of the social and cognitive components; P_{id}, pbest of particle i; P_{gd}, gbest of the group.

Adaptive inertia weight (AIW)

We use fallowing adaptive inertia weight (Nickabadi et al., 2008):

$W(t)=(W_{max}-W_{min})*P_s(t)+W_{min}$ (3)

W_{max}: initial weight, $W_{min:}$ final weight.

$$P_s(t)=\frac{\sum_{i=1}^{n} S(i,t)}{n} \qquad (4)$$

where n is number of particles and $P_s \in [0,1]$ is the percentage of the particles which have had an improvement in their fitness in the last iteration where the success of particle i at iteration t is defined as:

$$S\ (i,t)=\begin{cases} 1 & if \quad fit(pbest_i^t) < fit(pbest_i^{t-1}) \\ 0 & if \quad fit(pbest_i^t) = fit(pbest_i^{t-1}) \end{cases} \qquad (5)$$

where $pbest_i^t$ is the best position found by particle i until iteration t and fit () is the function to be optimized.

Problem formulation

The fitness of panicles is easily measured as the quantization error. The fitness function of the data clustering problem is given as follows:

$$f=\frac{\sum_{i=1}^{N_C}\left\{\frac{\sum_{j=1}^{p_i} d(0_i,m_{ij})}{p_i}\right\}}{N_c} \qquad (6)$$

The function f should be minimized. m_{ij}, jth data vector belongs to cluster i; O_i, Centroid vector of the ith cluster; $d(O_i, m_{ij})$, the distance between data vector m_{ij} and the cluster centroid O_i; P_i, the number of data set, which belongs to cluster Ci; N_c, number of clusters.

Particle representation

In the context of clustering, a single particle represents the cluster centroid vectors. That is, each particle X_{ij} , is constructed as follows:

$X_{ij}=(m_{i1}, m_{i2}\ldots m_{im})$

where, m_{ij} refers to the j-th cluster centroid vector of the i-th particle in cluster $m_{ij;}$ for initializing m_j a row of elements are chosen randomly from the matrix C.

$m_{ij}=(c_{j1}, c_{j2}, c_{j3} \ldots\ldots c_{jn})$

From the view of Figure 2, n is number of total texts and the dimension can be reduced from n to k(k<n).
$m_{ij}=(c_{j1}, c_{j2}, c_{j3}, \ldots., c_{jk})$

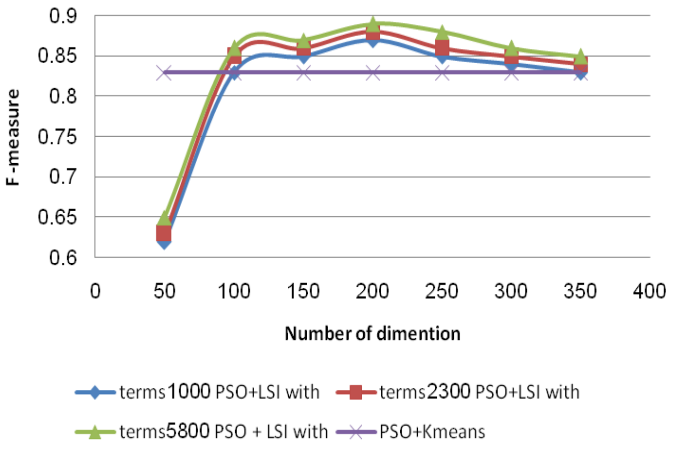

Figure 2. Cluster performance against the number of dimension on dataset 1.

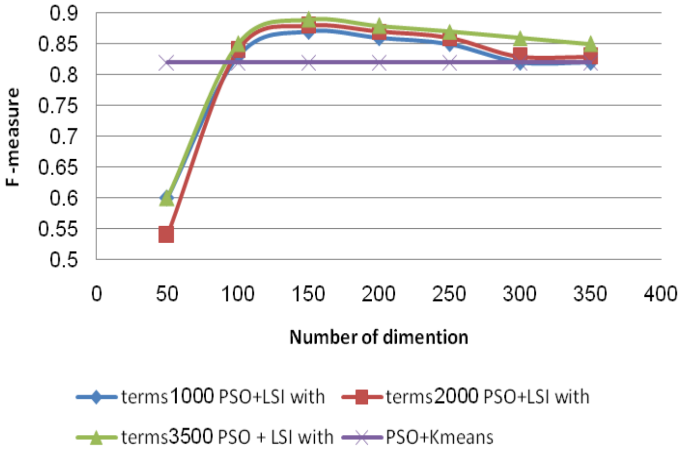

Figure 3. Cluster performance against the number of dimension on dataset 2.

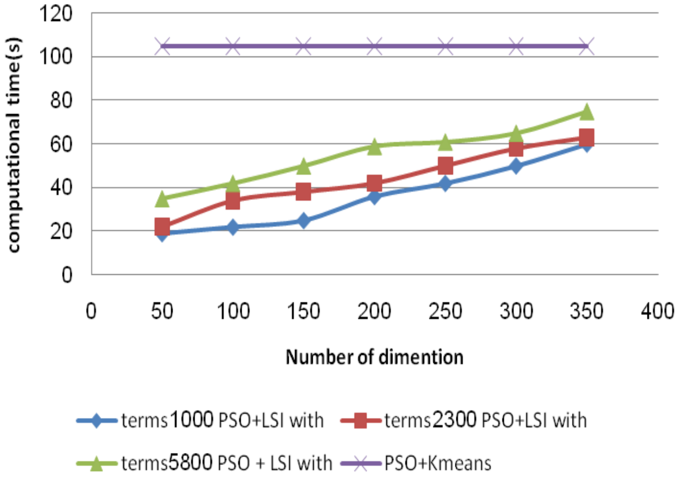

Figure 4. Computational time against the number of dimension on dataset 1.

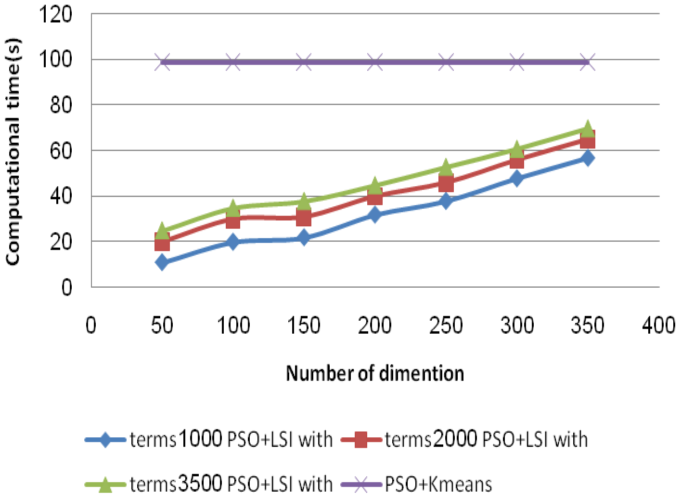

Figure 5. Computational time against the number of dimension on dataset 2.

EXPERIMENT

We present experiments in this section to demonstrate the effectiveness of the PSO+LSI algorithm. The experiments are performed on the benchmark dataset (Reuters (http://www.ics.uci.edu/~mlearn/MLRepository.htm.) and *Hamshahri* (Darrudi et al., 2004). Dataset 1 including 600 texts from Reuters and dataset 2 including 350 texts from Hamshahri dataset. After being processed by word extraction, stop word removal, and stemming, there are 5800 and 3500 terms, respectively. In our algorithm the weight of a term is measured by TF/IDF. However, the whole number of terms is not suitable for PSO, so we choose the terms with highest weight in the vocabulary. We varied the number of terms from 5800, 2300 to 1000 and 3500, 2000 to 1000 for dataset 1 and dataset 2 respectively to construct corpus matrix *C* in section 3. In the PSO clustering algorithm we choose 50 particles. In our experiments, it needed less than 400, 310 iterations for PSO+LSI algorithm to convert to the optimal result for dataset 1 and dataset 2 respectively. The *F*-measure (Song et al., 2006) is used for clustering evaluation.

RESULT

The cluster results are shown in Figures 2 and 3. The horizontal lines represent the cluster results in VSM with 5800 and 3500 features. Comparisons of the computational time are shown in Figures 4 and 5.

From Figure 2, we can see that from about 100 dimensions the performance of PSO+LSI outperforms that of PSO+Kmeans (Cui et al., 2005; Meijie et al., 2007) using VSM. For 5800 terms with LSI, PSO+LSI obtain its best performance on 200 dimensions. For 2300 and 1000 terms with LSI, PSO+LSI methods obtain their best performance on 200 dimensions. Furthermore, on the dimension of 200, the result of the PSO+LSI method with 1000 terms is very close to that with 2300 terms. From Figure 3, we can see that from about 100 dimensions, the performance of PSO+LSI outperform that of

PSO+Kmeans algorithms using the vector space model. When the dimension is 150, PSO+LSI methods with 1000, 2000 and 3500 terms obtain their best performance. Also, with 150 dimensions, the performance of 1000 terms with LSI is close to that of 2000 terms with LSI.

We can see from Figure 4 that the computational time of PSO+LSI is increased with a higher dimensionality. For all numbers of dimensions, PSO+LSI are faster than PSO+Kmeans algorithms using the vector space model. Furthermore, with 200 dimensions, it takes 36.3 s for the PSO+LSI method with 1000 terms to obtain its best performance, which is much faster than that on VSM model.

We can see from Figure 5 that for all numbers of dimensions, PSO+LSI are faster than PSO+Kmeans algorithms using VSM. It takes 22.3 s for PSO+LSI method with 1000 terms to obtain its best performance on 150 dimensions.

Conclusion

In this paper, we propose a method of PSO algorithm based on the latent semantic indexing model (PSO+LSI). Also the use adaptive inertia weight in PSO algorithm can cause successful exploration and exploitation in search space and fast convergence. Analyses shows that LSI not only provide an underlying semantic structure for text model but also reduces dimension drastically which is very suitable for PSO for evolving to optimal text cluster.

REFERENCES

Berry MW, Dumais ST, OBrien GW (1995). Using linear algebra for intelligent information retrieval. SIAM Rev., 37(4): 573-595.

Cui X, Potok T, Palathingal P (2005). Document Clustering Using Particle Swarm Optimization. Applied Software Engineering Research Group, Computational Sciences and Engineering Division, Oak Ridge National Laboratory, TN 37831-6085, IEEE.

Cagnina L, Errecalde M, Ingaramo D, Rosso P (2008). A iscrete Particle Swarm Optimizer for Clustering Short-text Corpora. In Proc. of the 3rd International Conference on Bioinspired Optimization Methodsand their Applications (BIOMA08), pp 93–103, Ljubljana, Slovenia.

Ding CH (2000). A probabilistic model for dimensionality reduction in information retrieval and filtering. In Proc. of 1st SIAM Computational Information Retrieval Workshop.

Duda RO, Hart PE (1993). Pattern Classification and Scene Analysis. John Wiley and Sons, USA. pp. 821–828.

Deerwester S, Dumais ST, Furnas GW, Landauer TK, Harshman R (1990). Indexing by latent semantic analysis. J. Am. Soc. Informat. Sci., 41(6): 391-407.

Darrudi E, Hejazi MR, Oroumchian F (2004). Assessment of a Modern Farsi Corpus. The Second Workshop on Information Technology and its Disciplines, WITID.

Errecalde M, Ingaramo D, Rosso P (2008). Proximity estimation and hardness of short-text corpora. In 5th International Workshop on Text-based Information Retrieval (TIR-2008), 19th International Conference on Database and Expert Systems Application, Dexa, pp. 15-19, IEEE Computer Society.

Ercan MF (2008). A hybrid particle swarm optimization approach for scheduling flow-shops with multiprocessor tasks. Int. Confer. Inf. Sci. Sec.. pp. 6922-6924.

Evangelou IE, Hadjimitsis DG, Lazakidou AA, Clayton C (2001). Data Mining and Knowledge Discovery in Complex Image Data using Artificial Neural Networks. Workshop on Complex Reasoning an Geographical Data, Cyprus.

Fukunaga K (1990). Introduction to Statistical Pattern Recognition". Academic Press. 006.4 F955i.

Gareth J, Robertson AM, Santimetvirul C, Willett P (1995). Non-hierarchic document clustering using a genetic algorithm. Inf. Res.,

Kennedy J, Eberhart RC, Shi Y (2001). Swarm Intelligence. Morgan Kaufmann, NewYork. pp. 303–308.

Lillesand T, Keifer R (1994). Remote Sensing and Image Interpretation. John Wiley & Sons, USA. pp. 239-255.

Meijie ZHU, Hanxing LIU, Weiwei SUN, TongLin ZHU (2007). Improvement of Particle Swarm Optimization Based on Neighborhood Cognizance and Swarm Decision. IEEE.

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

Nickabadi A, Ebadzadeh M, Safabaksh R (2008). Particle swarm optimization algorithm with adaptive inertia weight: a survey of the state of the art and a novel method. IEEE Trans. Evolutionary Comput., pp.86-92.

Passaro A, Starita A (2008). Particle Swarm Optimization for Multimodal Functions: A Clustering Approach. In Journal of Artificial Evolution and Applications, Article ID 482032, 15doi: 10: 1155.

Pinto D, Salazar HJ, Rosso P (2006). Clustering Abstracts of Scientific Texts Using the Transition Point Technique. In Proc. of the CICLing 2006 Conference, LNCS, Springer-Verlag, 3878: 536–546.

Raghavan VV, Birchand K (1999). A clustering strategy based on a formalism of the reproductive process in a natural system. Proceedings of the Second International Conference on Information Storage and Retrieval, pp. 10–22.

Rao MR (1991). Cluster Analysis and Mathematical Programming. Journal of the American Statistical Association, 22: 622-626.

Sorg P, Cimiano P (2008). Cross lingual Information Retrieval with Explicit Semantic Analysis. In Working Notes of the Annual CLEF Meeting.

Sneath PH, Sokal RR (1973). Numerical Taxonomy. Freeman, London, UK. p. 359.

Song W, Park SC (2006). Genetic algorithm-based text clustering technique. *LNCS,* 4221: 779–782.

UCI Repository for Machine Learning Databases retrieved from the World Wide Web: http://www.ics.uci.edu/~mlearn/MLRepository.htm.