DE GRUYTER OPEN

JAISCR, 2015, Vol. 5, No. 4, pp. 231 – 238
10.1515/jaiscr-2015-0031

FEATURE SELECTION USING PARTICLE SWARM OPTIMIZATION IN TEXT CATEGORIZATION

Mehdi Hosseinzadeh Aghdam[1] and Setareh Heidari[2]

[1]*Department of Computer Engineering and Information Technology, Payame Noor University (PNU)P.O.BOX 19395-3697, Tehran, IRAN*

[2] *School of Computer Engineering Iran University of Science and Technology Tehran, IRAN*

Abstract

Feature selection is the main step in classification systems, a procedure that selects a subset from original features. Feature selection is one of major challenges in text categorization. The high dimensionality of feature space increases the complexity of text categorization process, because it plays a key role in this process. This paper presents a novel feature selection method based on particle swarm optimization to improve the performance of text categorization. Particle swarm optimization inspired by social behavior of fish schooling or bird flocking. The complexity of the proposed method is very low due to application of a simple classifier. The performance of the proposed method is compared with performance of other methods on the Reuters-21578 data set. Experimental results display the superiority of the proposed method.

1 Introduction

Feature selection (FS) is used in many areas as a tool to eliminate irrelevant and redundant features. FS simplifies a data set by reducing its dimensionality and identifying relevant features without decreasing the prediction accuracy. The dimensionality of data set are often very large, since learning algorithm might not work as well before removing these irrelevant features. Reducing the number of irrelevant features significantly reduces the running time of a learning algorithm. FS has many applications, including text categorization (TC), data mining, pattern recognition and signal processing [1].

The goal of TC is automatically assigning predefined categories to text documents [2]. This goal is of great practical importance given the enormous bulk of online text available through the web sites, emails, and digital libraries. A main challenge of TC is the high dimensionality of the feature space. The original feature space contains of many unique terms that occur in texts, and the number of terms can be hundreds of thousands for even a moderate-sized text collection. This is highly costly for many mining methods. Thus, it is highly desirable to reduce the feature space without decreasing categorization accuracy.

Several methods have been applied to the problem of FS in TC. Yang and Pedersen conducted a comparative study on five FS criteria for TC, including document frequency, information gain, mutual information, a $\chi 2$-test (CHI) and term strength and found $\chi 2$ statistics and information gain more effective for optimizing classification results [3]. Kim, et al. examined three methods consist of centroid, orthogonal centroid, and LDA/GSVD, which are designed for reducing the dimension of clustered data for dimensional reduction in TC [4]. Forman presented an excellent review of FS methods for TC and introduced a case study in text feature selection [5].

Exhaustive search is the easiest way to determine the optimal subset of features by evaluating all the subsets. This method is quite impractical even for a medium size feature set. FS methods usually involve heuristic or random search strategies to avoid this complexity. However, the optimality of the final feature subset is often reduced [1].

Among many methods which are proposed for FS, heuristic methods such as genetic algorithm [6], ant colony optimization [7] and particle swarm optimization have been interested for researchers. These methods try to gather better solutions by using knowledge from previous steps. GAs are optimization methods based on the natural selection. They applied operations found in natural genetics to guide search in the search space [8]. Because of their advantages, GAs have been widely used as a tool for FS in data mining [9]. Particle swarm optimization which is introduced by Kennedy and Eberhart, is based on a social-psychological model of social influence and social learning. Particles in a swarm follow a very simple behavior: emulate the success of neighboring individuals. The emerged group behavior discovers optimal regions of a high dimensional search space.

In this paper a modified PSO-based FS method has been presented. The classifier performance and the length of selected feature subset are used as heuristic information for the proposed PSO-based method. Thus, the proposed method needs no prior knowledge of features. The proposed method is applied to text features of bag of words model in which a document is considered as a set of words or phrases and each position in the feature vector corresponds to a given term in document [10]. Finally, the length of selected feature vector and the classifier performance are considered for performance evaluation.

The rest of this paper is structured as follows. Section 2 presents a brief overview of FS methods. The proposed PSO-based feature selection algorithm is described in section 3. Section 4 reports computational experiments. It also includes a brief discussion of the results which are obtained and finally the conclusion and future works are offered in the last section.

2 Feature Selection Approaches

FS is a procedure that chooses a subset from the feature set. The optimality of a feature subset is evaluated by criterion. Since FS is a NP-hard problem, there is no practical solution to find its optimal feature subset [11]. A typical FS procedure contains subset generation, subset evaluation, termination criteria and result validation [12]. Subset selection process implements a search method that chooses feature subsets for evaluation based on a certain search method. These search methods includes forward selection, backward elimination and forward/backward combination methods. The process of subset selection and evaluation is repeated until a given termination condition is satisfied. The selected best feature subset usually needs to be validated using a different test data set [13]. The methods to feature subset selection can be categorized into filters, wrappers and embedded approaches. The filter model separates FS from classifier learning and selects feature subsets that are independent of any learning algorithm [1]. In the wrapper method feature subset is chosen using the evaluation function based on the same learning algorithm that will be used later for learning. In this method the evaluation function computes the suitability of a feature subset generated by the subset generation procedure and it also compares that with the previous best candidate, replacing it if found to be better. A stopping criterion is tested in each of iterations to determine whether or not the FS procedure should continue. Although, wrappers may generate better solutions, they have complexity to run and can break down with very large numbers of features since they use of learning algorithms in the evaluation of subsets. If the FS and learning algorithm are interleaved then the FS method is a type of embedded method [14].

3 Particle Swarm Optimization for Feature Selection

The PSO is a computational approach that optimizes a problem in continuous, multidimensional search spaces. PSO starts with a swarm of random particles. Each particle is associated with a velocity. Particles' velocities are adjusted in order to the historical behavior of each particle and its neighbors

during they fly through the search space. Thus, the particles have a tendency to move towards the better search space [15]. The version of the utilized PSO algorithm is decribed mathematically by the following equations:

$$V_i(t+1) = w.V_i(t) + c_1.r_1(t).[P_i(t) - X_i(t)] + c_2.r_2(t).[P_g(t) - X_i(t)] \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

where c_1 and c_2 are positive constants, called learning rates; $P_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ shows the best previous position of the swarm; r_1 and r_2 are random values in the range [0, 1]; $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ displays the position of the ith particle in a problem space with D dimensions; t indicates the iteration number; w is a inertia weight; the index g represents the best particle among all the particles in the population; $V_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$ indicates the rate of change of position (velocity). If the sum of the factors in the right side of equation (1) exceeds a specified constant value, particles' velocities on each dimension are clamped to a maximum velocity V_{max} [15].

PSO was originally considered for searching multidimensional continuous spaces. In this paper, it is adapted to the discrete FS problem. Each feature subset can be considered as a point in feature space. The optimal point is the subset with least length and highest classification accuracy. The initial swarm is distributed randomly over the search space, each particle takes one position. The goal of particles is to fly to the best position. By passing the time, their position are changed by communicating with each other, and they search around the local best and global best position. Finally, they should converge on good, possibly optimal, positions since they have exploration ability that equip them to perform FS and discover optimal subsets.

PSO needs to be extended in order to deal with FS. The particle's position is considered as binary bit strings. Every bit represents a feature; the bit value 1 represents a selected feature, whereas the bit value 0 represents a non-selected feature. Each position is a feature subset. To apply the PSO idea to optimization problem, the following problem-dependent aspects can be defined.

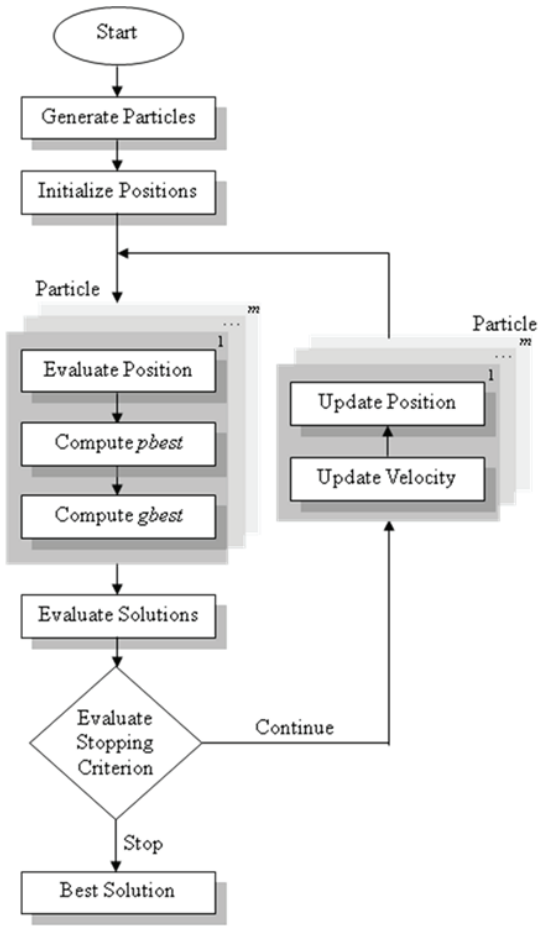

Figure 1. PSO-based feature selection algorithm

3.1 Updating velocity

The velocity of each particle is displayed as a positive integer; particle velocities are bounded to a maximum velocity V_{max}. It shows how many of features should be changed to be same as the global best point, in other words, the velocity of the particle moving toward the best position. The number of different features (bits) between two particles related to the difference between their positions. For example, P_g= [1 0 1 1 1 0 1 0 0 1], X_i = [0 1 0 0 1 1 0 1 0 1]. The difference between P_g and the particle's current position is P_g - X_i = [1 -1 1 1 0 -1 1 -1 0 0]. A value of 1 indicates that compared with the best position, this feature should be selected but is not, which will decrease classification quality and lead to a lower fitness value. Assume that the number of 1's is M. On the other hand, a value of -1 indicates that, compared with the best position, this bit should not be selected, but is selected. Redundant features will make the length of the subset longer and lead

to a lower fitness value. The number of -1's is N. We use the value of (M+N) to express the distance between two positions; (M+N) may be positive or negative. Such variation makes particles exhibit exploration ability within the solution space. In this example, (M+N) = 4 + 3 = 7, so P_g - X_i = 7.

3.2 Updating position

After updating the velocity, a particle's position will be updated by the new velocity. Suppose that the new velocity is V. In this case, V bits of the particle are randomly changed, different from that of P_g. The particles then fly toward the global best while still exploring the search area, instead of simply being same as P_g.

The V_{max} is used as a constraint to control the global exploration ability of particles. A larger V_{max} provides global exploration, while a smaller V_{max} increases local exploitation. When V_{max} is low, particles have difficulty getting out from locally optimal sections. If V_{max} is too high, swarm might fly past good solutions [16]. We set V_{max} = (1/2)N and limit the velocity within the range [1, (1/2)N], which prevents an overly-large velocity. A particle can be close to an optimal solution, but a high velocity may make it move far away. By limiting V_{max}, particles cannot move too far away from the optimal solution.

3.3 Fitness function

The fitness function is defined in equation (3):

$$Fitness(X_i) = \phi.\gamma(S^i(t)) + \varphi.(n - |S^i(t)|) \qquad (3)$$

where $S^i(t)$ is the feature subset found by particle i at iteration t, and $|S^i(t)|$is its length. Fitness is computed in order to both the measure of the classifier performance,$\gamma(S^i(t))$, and feature subset length. ϕ and φ are two parameters that control the relative weight of classifier performance and feature subset length, $\phi \in [0,1]$ and $\varphi = 1 - \phi$. This formula denotes that the classifier performance and feature subset length have different effect on FS. This paper considers that classifier performance is more important than subset length, so we set them to ϕ=0.8, φ=0.2.

3.4 Solution construction

The overall process of PSO for feature selection can be seen in Figure 1. The process begins by generating a number of particles which are then placed randomly on the search space, i.e. each particle starts with one random position. Alternatively, the number of particles to place on the search space may be set equal to the number of features within the data; each particle starts finding solution by a movement. From these initial positions, they fly through solutions in each iteration. The selected subsets are collected and then evaluated. If a best subset has been discovered or the algorithm has run a certain number of times, then the process stops and returns the best feature subset encountered. If none of these conditions are met, then the velocities are updated, the particles' positions are calculated and the process iterates once more.

4 Experimental Results

Table 1. Number of Training/Test Documents

Category Name	***Number of Train Documents***	***Number of Test Documents***
Acquisition	1484	664
Corn	170	53
Crude	288	126
Earn	2721	1052
Grain	72	32
Interest	165	74
Money-fx	313	106
Ship	122	42
Trade	297	99
Wheat	153	51

A series of experiments was conducted to show the utility of proposed FS algorithm. All experiments are executed on a machine with Intel(R) Core(TM) i7 CPU 3.2 GHz and 4 GB of RAM. We implement the proposed PSO algorithm and other three FS algorithms in Matlab. The operating system was Windows 7 Professional. We used Reuters-21578, the newer version of the corpus [17]. In Reuters-21578 data set, we select the top ten classes; 5785 documents in training set and 2299 documents in test set. The distribution of the class is unbalance. The maximum class has

2721 documents, occupying 47.04 % of training set. The minimum class has 72 documents, occupying 1.24% of training set. Table 1 presents the ten most frequent categories.

4.1 Feature Extraction

Text documents cannot be directly interpreted by a classifier. So, an indexing procedure that maps a text document into a compact representation of its content needs to be uniformly applied to documents. Therefore, a text d_j is usually represented as a vector of term weights. Typically each position in the input feature vector equals to a given word or phrase. This representation often called *bag of words* model. Weights are determined using normalized *tfidf* function [18], defined as:

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|}(tfidf(t_s, d_j))^2}} \tag{4}$$

where is the set of terms (features) that occur at least once in at least one document of training set, and $0 \leq w_{kj} \leq 1$ represents, how much term t_k contributes to the semantics of document d_j.

$$tfidf(t_k, d_j) = \#(t_k, d_j).\log\frac{|Tr|}{\#_{Tr}(t_k)}. \tag{5}$$

where #(t_k,d_j) is the number of occurrence of t_k in d_j, Tr is the training set and $|Tr|$is its length. $\#_{Tr}(t_k)$ denote the number of documents in Tr in which t_k occurs.

4.2 Performance Measure

Typically precision (π) and recall (ρ) are used for measurement. They showed in equations (6) and (7).

$$\pi_i = \frac{TP_i}{TP_i + FP_i} \tag{6}$$

$$\rho_i = \frac{TP_i}{TP_i + FN_i} \tag{7}$$

where TP_i is the number of test documents correctly classified under i-th category (c_i), FP_i is the number of test documents incorrectly classified under c_i, and FN_i is the number of test documents incorrectly classified under other categories these probabilities may be estimated in terms of the contingency Table for c_i on a given test set. Another commonly used measure in TC is *F1* measure that is defined in equation (8) [19] [20].

Table 2. The Global Contingency Table

Category set $C = \{c_1,\ldots,c_{\|C\|}\}$		**Expert judgments**	
		Yes	***No***
Classifier	***Yes***	TP = $\sum_{i=1}^{\|C\|} TP_i$	FP = $\sum_{i=1}^{\|C\|} FP_i$
Judgments	***No***	FN = $\sum_{i=1}^{\|C\|} FN_i$	TN = $\sum_{i=1}^{\|C\|} TN_i$

$$F1 = \frac{2 \times \pi \times \rho}{(\pi + \rho)} \tag{8}$$

When facing multiple classes there are two possible ways of averaging above measures, namely, macro average and micro average. The macro average weights equally all the classes, regardless of how many documents belong to it. The micro average weights equally all the documents, thus favoring the performance on common classes. The *global* contingency table which is shown in Table 2 is thus obtained by summing over category-specific contingency tables; equations (9) to (12) show micro averaging and macro averaging on precision and recall.

$$\pi^{\mu} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} \tag{9}$$

$$\rho^{\mu} = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FN_i)} \tag{10}$$

$$\pi^{\mathrm{M}} = \frac{\sum_{i=1}^{|C|} \pi_i}{|C|} \tag{11}$$

$$\rho^{\mathrm{M}} = \frac{\sum_{i=1}^{|C|} \rho_i}{|C|} \tag{12}$$

where μ denotes micro averaging and denotes macro averaging.

4.3 Results

To show the utility of proposed PSO-based algorithm we compare the proposed algorithm with genetic algorithm, information gain and CHI. Various values were tested for the parameters of proposed algorithm. The results show that the highest performance is achieved by setting the parameters to values as follow:

The population size is 50, the maximum number of iteration is 100, C_1=C_2=1 and w is in the range of [0.4, 1.4]. These values were empirically determined in our preliminary experiments; but we

make no claim that these are optimal values. Parameter optimization is a topic for future research.

Analyzing the precision and recall shown in Table 3, on average, the PSO-based algorithm obtained a higher accuracy value than the genetic algorithm, information gain and CHI. To graphically illustrate the progress of the PSO as it searches for optimal solutions, we take percent features as the horizontal coordinate and the *F1* measure as the vertical coordinate. This should illustrate the process of improvement of the best particle as the number of features increase.

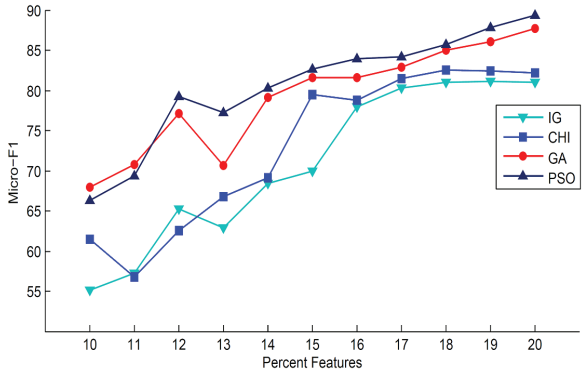

Figure 2. Comparison of micro-F1 of four algorithms

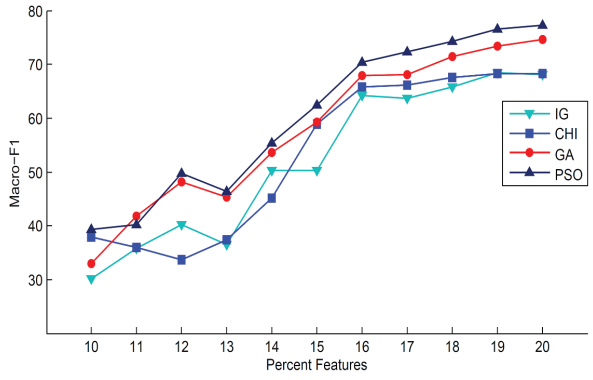

Figure 3. Comparison of macro-F1 of four algorithms

Figures 2 and 3 show the micro-averaged and macro-averaged F1 measure for each of the FS algorithms as we change the number of selected features. The results display that as the percentage of selected features exceeds 12% in micro-F1 and macro-F1 measures, the PSO-based algorithm outperforms genetic algorithm, information gain and CHI.

Table 4. Micro-F1 and Macro-F1 of Four Algorithms

Feature Selection Algorithms	***Macro-F1***	***Micro-F1***
IG	69.8124	80.9482
CHI	70.8601	82.2097
GA	76.2685	86.3854
PSO	78.8564	89.5684

Table 4 describes micro-F1 and macro-F1 for four FS algorithms. From this Table, the best categorization performance is achieved with GA and PSO. Compared with GA, PSO is quicker in locating the optimal solution. In general, it can find the solution within tens of iterations. If exhaustive search is used to find the optimal feature subset in the Reuters-21578 data set, there will be tens of billions of candidate subsets, which is impossible to execute. But with PSO, at the 100nd iteration the solution is found.

5 Conclusion

Exhaustive searches are impossible for even medium sized data sets. Thus, stochastic methods provide a promising FS mechanism. This paper proposes a FS technique based on particle swarm optimization. We compare its performance with other FS methods in TC. PSO has the ability to converge quickly; it has a strong search capability on the problem space and can efficiently find minimal feature subset.

In the proposed algorithm, the classifier performance and the length of selected feature subset are adopted as heuristic information. So, we can select the best feature subset without any prior knowledge of features. To show the utility of the proposed algorithm and to compare it with information gain and CHI, a set of experiments were carried out on Reuters-21578 data set. The computational results indicate that proposed algorithm outperforms information gain and CHI methods since it achieved better performance with the lower number of features. To show the effectiveness of the proposed algorithm, we have used a simple classifier (nearest neighbor classifier) which can affect

Table 3. The Performance (Precision and Recall) of Information Gain, CHI, GA and PSO on Reuters-21578 Data set

Category Name	IG		CHI		GA		PSO	
	Precision	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*	*Precision*	*Recall*
Acquisition	91.7466	71.988	89.5575	76.2048	92.4528	81.1747	91.5264	90.5648
Corn	87.8049	67.9245	87.8049	67.9245	90.6977	73.5849	91.6321	74.2145
Crude	84.9057	71.4286	83.0189	69.8413	78.3217	88.8889	81.6942	88.8889
Earn	84.3803	94.4867	84.9829	94.6768	90.1961	96.1977	97.5612	94.3328
Grain	63.6364	65.625	60.6061	62.5	69.697	71.875	67.264	74.3568
Interest	45.7627	36.4865	54	36.4865	60	36.4865	50.6841	52.6428
Money-fx	51.7241	56.6038	61.2245	56.6038	67.8899	69.8113	68.4863	70.8113
Ship	33.8235	54.7419	37.5	57.1429	57.1429	66.6667	60.3521	75.3621
Trade	68.7023	90.9091	73.9837	91.9192	72.3577	89.899	78.6325	91.8751
Wheat	91.3043	82.3529	89.3617	82.3529	87.7551	84.3137	86.7864	85.3621
Average	70.3791	69.2547	72.204	69.5653	76.6511	75.8898	77.4619	79.8411

the categorization performance. As for the future work, intention is to apply the proposed FS algorithm using complicated classifiers to improve its performance and to combine the proposed method with other population-based algorithms.

References

[1] Jensen, R. (2005). Combining rough and fuzzy sets for feature selection. Ph.D. dissertation, School of Information, Edinburgh University.

[2] W. Shang, H. Huang, H. Zhu, Y. Lin, Y. Qu, and Z. Wang, A novel feature selection algorithm for text categorization, Expert Systems with Applications, vol. 33(1), pp. 1-5, 2007.

[3] Y. Yang, and J.O. Pedersen, A Comparative Study on Feature Selection in Text Categorization, Proceedings of the 14th International Conference on Machine Learning, pp. 412-420, 1997.

[4] H. Kim, P. Howland, and H. Park, Dimension Reduction in Text Classification with Support Vector Machines, Journal of Machine Learning Research, 6, 37–53, 2005.

[5] G. Forman, Feature Selection for Text Classification, In: Computational Methods of Feature Selection, Chapman and Hall/CRC Press, 2007.

[6] M. Raymer, W. Punch, E. Goodman, L. Kuhn, and A.K. Jain, Dimensionality Reduction Using Genetic Algorithms, IEEE Transactions on Evolutionary Computing, 4, pp. 164–171, 2000.

[7] M.H. Aghdam, N. Ghasem-Aghaee, and M.E. Basiri, Application of ant colony optimization for feature selection in text categorization, Proceedings of the IEEE Congress on Evolutionary Computation, pp. 2872-2878, 1-6, 2008.

[8] M. Srinivas, L.M. and L.M. Patnik, Genetic Algorithms: A Survey, IEEE Computer Society Press, Los lamitos, 1994.

[9] W. Siedlecki, and J. Sklansky, A note on genetic algorithms for large-scale feature selection, Pattern Recognition Letters, vol. 10(5), pp. 335-347, 1989.

[10] M.F. Caropreso, and S. Matwin, Beyond the Bag of Words: A Text Representation for Sentence Selection, StateplaceBerlin: Springer-Verlag, pp. 324-335, 2006.

[11] R. Kohavi, and G.H. John, Wrappers for feature subset selection, Journal of Artificial Intelligence, vol. 97(1-2), pp. 273–324, 1997.

[12] M. Dash, and H. Liu, Feature selection for classification, Intelligent Data Analysis: An International Journal, vol. 1(3), pp. 131-156, 1997.

[13] H. Liu, and L. Yu, Toward Integrating Feature Selection Algorithms for Classification and Clustering, IEEE Transactions on Knowledge and Data Engineering, vol. 17(4), pp. 491-502, 2005.

[14] D. Mladeni, Feature Selection for Dimensionality Reduction. Subspace, Latent Structure and Feature Selection, Statistical and Optimization, Perspectives Workshop, SLSFS 2005, CityplaceBohinj, country-regionSlovenia, Lecture Notes in Computer Science 3940 Springer, pp. 84–102, 2006.

[15] J. Kennedy, R.C. Eberhart, Particle swarm optimization, Proceedings of IEEE International Conference on Neural Networks, pp. 1942-1948, 1995.

[16] A.P. Engelbrecht, Fundamentals of Computational Swarm Intelligence, John Wiley & Sons, London, 2005.

[17] The reuters-21578 text categorization test collection. Available: http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

[18] G. Salton, and C. Buckley, Term-weighting approaches in automatic text retrieval, PlaceNameCornell PlaceTypeUniversity CityplaceIthaca, StateNY, country-regionUSA, Technical Report TR87-881, 1987.

[19] C.J. Rijsbergen, Information Retrieval, 2nd ed. Butterworths, London, UK, 1979.

[20] M.H. Aghdam, J. Tanha, A.R. Naghsh-Nilchi, and M.E. Basiri, Combination of Ant Colony Optimization and Bayesian Classification for Feature Selection in a Bioinformatics Dataset, Journal of Computer Science & Systems Biology vol. 2, pp. 186-199, 2009.

Mehdi Hosseinzadeh Aghdam is a Ph.D. candidate in the computer engineering department at the Iran University of Science and Technology, also he is a lecturer at Payame Noor University. He graduated with a master's in computer engineering Artificial Intelligence from University of Isfahan (UI) in 2008. At UI, he worked on swarm intelligence-based method for feature selection. His main research interests are: Data Mining, Computational Intelligence, and Pattern Recognition.

Setareh Heidari has graduated with a master in computer networks information technology in the Iran University of Science and Technology. At IUST, She worked on Formal Verification of security protocol. Her main research interests are: Swarm Intelligence, Network Security, Wireless Networks and Sensor Networks.