Academic Platform Journal of Engineering and Science 6-2, 134-141, 2018

Academic Platform Journal of Engineering and Science

journal homepage: http://apjes.com/

Parçacık Sürüsü Eniyilemesine Dayalı Yığılmış Genelleme Yöntemi ve Metin Sınıflandırma Üzerinde Uygulanması

*[1]Aytuğ Onan(0000-0002-9434-5880)

[1]Manisa Celal Bayar Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, Manisa
aytug.onan@cbu.edu.tr

Arrival Date: 20.07.2017 Accepted Date: 31.05.2018

Öz

Topluluk öğrenmesi, birden fazla öğrenme algoritmasının çıktılarının birleştirilmesi ile daha yüksek başarımlı ve güvenilir sınıflandırma modelleri oluşturulmasını amaçlar. Topluluk öğrenmesi yöntemleri, aralarında metin madenciliğinin de yer aldığı birçok alanda başarı ile uygulanmaktadır. Yığılmış genelleme algoritması, heterojen sınıflandırma algoritmaları ile sınıflandırıcı topluluğu oluşturulmasına yönelik bir yöntemdir. Yığılmış genelleme algoritmasında, temel öğrenme algoritmalarının çıktıları, üst seviyeli bir öğrenme algoritması aracılığıyla birleştirilir. Yığılmış genelleme algoritmasının etkin bir biçimde işleyebilmesi için, temel öğrenme algoritması olarak görev alacak yöntemlerin seçilmesi gerekmektedir. Bunun yanı sıra, üst seviye öğrenme algoritması olarak hangi yöntemin kullanılacağının belirlenmesi gereklidir. Bu nedenle, yığılmış genelleme algoritması için uygun bir konfigürasyon belirlenmesi, zor bir problemdir. Bu çalışmada, yığılmış genelleme algoritması için uygun bir konfigürasyon belirlenmesi işlemi bir eniyileme problemi olarak ele alınmış ve parçacık sürüsü eniyilemesine dayalı bir yöntem önerisinde bulunulmuştur. Metin sınıflandırma alanında gerçekleştirilen deneysel analizlerde, parçacık sürüsü eniyilemesine dayalı yöntem, genetik algoritma, karınca kolonisi eniyilemesi ve yapay arı kolonisine dayalı yığılmış genelleme yöntemleri ile karşılaştırılmıştır.

Anahtar Kelimeler: Sınıflandırıcı topluluğu, yığılmış genelleme yöntemi, parçacık sürüsü eniyilemesi, metin sınıflandırma.

Particle Swarm Optimization Based Stacking Method with an Application to Text Classification

*[1]Aytuğ Onan

[1]Manisa Celal Bayar University, Faculty of Technology, Department of Software Engineering, Manisa
aytug.onan@cbu.edu.tr

Abstract

Multiple classifier aims to integrate the predictions of several learners so that classification models can be constructed with high performance of classification. Multiple classifiers can be employed in several application fields, including text categorization. Stacking is an ensemble algorithm to construct ensembles with heterogeneous classifiers. In Stacking, the predictions of base-level classifiers are integrated by a meta-learner. To configure Stacking, appropriate set of learning algorithms should be selected as base-level classifiers. Besides, the learning algorithm that will perform the meta-learning task should be identified. Hence, the identification of an appropriate configuration for Stacking can be a challenging problem. In this paper, we introduce an efficient method for stacking ensemble based text categorization which utilizes particle swarm optimization to upgrade arrangement of the ensemble. In the empirical analysis on text categorization domain, particle swarm optimization based Stacking method has been compared to genetic algorithm, ant colony optimization and artificial bee colony algorithm.

Keywords: Classifier ensembles, stacking method, particle swarm optimization, text classification.

1. INTRODUCTION

Multiple classifier system (also known as ensemble learning) is a promising field in pattern recognition. The principle thought behind multiple classifier is to obtain a combined prediction model with higher predictive performance based on multiple classifiers. Multiple classifier can result in reduction of the variance of learners.

*Corresponding Author: Manisa Celal Bayar Üniversitesi, Teknoloji Fakültesi, Yazılım Mühendisliği Bölümü, manisaaytug.onan@cbu.edu.tr

Doi: 10.21541/apjes.329940

In addition, more expressive classification models can be generated [1]. Multiple classifier systems yield better classification accuracy than a single learning algorithm, due to representational, statistical and computational factors [2]. Statistically, with the existence of sufficient data, different classifier can be generated by sampling distributions. Computationally, the parameter dependency of learning algorithms and the risk of generating a local optimum result can be reduced. Besides, obtaining different representational configurations may be beneficial in some applications.

Multiple classifiers can be assigned into four groups based on the ways of building ensembles as data-level, feature-level, classifier-level and combination-level ensembles [1]. In data-level ensembles, different datasets obtained by bootstrapping of training dataset are trained on different classifiers. In feature-level, different feature subsets are utilized, whereas classifier-level ensembles use different base learning algorithms. In combination-level ensembles, different combiners are designed. Another taxonomy for classification of multiple classifiers is the structure of ensemble construction [3]. Based on the structure employed in ensemble generation from base learning algorithms, there are two main groups of ensemble algorithms, namely dependent algorithms and independent algorithms. In dependent algorithms, prediction of a classifier is employed to obtain the output of the subsequent classifier, whereas the output of classifiers are obtained separately and their results are integrated in independent methods [3]. There are several dependent learning methods, such as incremental batch learning (IBL) and model-guided instance selection (MGIS) [4]. In IBL, predictions obtained in a particular cycle is provided for the learner of ensuing cycle as a prior knowledge. Likewise, in model-guided instance selection, the learners of the earlier cycles are employed to manipulate the training set of ensuing cycles. Dependent algorithms include AdaBoost. In independent methods, several datasets are generated from the original dataset and these datasets are employed to train the classifiers. Then, the output for final classification is obtained by a combination method. There are several independent ensemble algorithms, such as Bagging and Random Forest method.

Combining base learning algorithms is another important issue in multiple classifier. The outputs of learners can be integrated by weighting methods and meta-learning algorithms [5]. The conventional weighting methods include majority voting, performance weighting, distribution summation, Bayesian combination and the conventional meta-learning algorithms include Stacking, arbiter trees, combiner trees and Grading [3].

Stacking (Stacked generalization) is a meta-learning based ensemble algorithm to combine multiple classification models [6]. Compared to the other multiple classifiers, such as Bagging and Boosting, Stacking has been less widely utilized in the literature [7]. Yet, the classification accuracy of learning algorithms can be substantially enhanced with the use of stacking algorithm [8]. In Stacking, the training set is divided into two disjoint set (namely, one training set and one test set).Based on the predictions of base learning algorithms and the correct responses, a meta-learner is trained [8]. To use stacking, there are a number of issues to be considered. The algorithms that will be employed as base-level classifiers and their parameters should be identified, the number of base learners should be determined, the algorithm that will be employed as a meta-learner and its parameters should be determined and types of features that will be employed to generate meta-data should be decided [9]. Hence, the identification of an appropriate/optimal configuration for Stacking is a challenging task.

Metaheuristic algorithms are widely recognized as established and efficient methods in optimization problems. Metaheuristic methods can be classified as single-solution metaheuristics and population-based metaheuristics [10]. Tabu search, simulated annealing and local search are some representatives of single-solution based metaheuristics and genetic algorithms, particle swarm optimization, ant colony optimization, artificial bee colony are population based metaheuristics. Metaheuristic algorithms have been successfully applied in a wide range of optimization problems, such as job shop scheduling, vehicle routing, resource allocation, pattern recognition, data mining, clustering and engineering design optimization [11, 12]. Some researchers introduced various metaheuristics to identify the optimal scheme for Stacking algorithm. In [13], a good configuration for Stacking algorithm is obtained by a genetic search algorithm. In [14], ant colony optimization is utilized to identify an optimized configurations for Stacking algorithm. In [15], artificial bee colony algorithm based method is proposed to configure Stacking algorithm. In this paper, we present a Stacking based scheme which employs particle swarm optimization algorithm to find an optimal configuration for base-level and meta-level classification algorithms.

2. THEORETICAL FOUNDATIONS

In this section, classification algorithms and multiple classification algorithms are introduced.

2.1. Classification Algorithms

In the experiments, eight machine learning algorithms (namely, logistic regression, Naïve Bayes algorithm, C4.5, K-nearest neighbour algorithm, K-star algorithm, ZeroR algorithm, Decision Stump algorithm and PART algorithm) have been considered.

Logistic regression (LR) is a linear learner, which employs a linear function to estimate the class labels for each instance [16]. Linear regression algorithm can be utilized for classification and regression problems. LR algorithm can be employed for classification problems with nominal features.

Naïve Bayes algorithm (NB) is a probabilistic learner, which employs Bayes' theorem with independence assumptions between attributes [17]. Though the algorithm is based on the independence assumption, the algorithm is scalable with comparable predictive performance to other conventional learning algorithms, such as k-nearest neighbor algorithm and support vector machines. NB algorithm is regarded as a standard technique for many pattern recognition tasks, including text categorization.

C4.5 algorithm is a decision tree based learner, which is a successor of ID3 [18]. In the algorithm, the selection of test feature is determined based on the information gain metric. For a particular set, the algorithm identifies a feature with the highest information gain as the attribute. The algorithm has a pruning mechanism. Hence, the algorithm can eliminate overfitting and can deal with noisy instances [19].

K-nearest neighbour algorithm (KNN) is a nonparametric lazy learner, which can be employed for classification and regression problems [20]. Regarding classification problem, the class label for a new instance is determined by majority voting of class labels for *k*-closest instances.

K-star algorithm is another instance based learner, which employs entropy-based evaluation function to estimate the class label of each instance. The algorithm can be employed for problems with symbolic and real-valued feature sets.
ZeroR algorithm is a simple classifier that relies on the target value and ignores all the predictors [15]. The algorithm simply predicts the majority category (class).

Decision Stump (DS) is a decision tree based learner [22]. In this scheme, one-level decision tree is constructed, such that there is one root and its leaves. The class label for a particular instance is determined based on the value of a single attribute.

PART algorithm is a rule based learner, which builds partial decision trees to obtain classification rules [23]. In this scheme, decision trees are constructed with the use of C4.5 algorithm. In each iteration of algorithm, decision tree returns the best leaf as a classification rule.

2.2. Multiple Classifier Methods

In the empirical analysis, four multiple classifier methods (namely, Bagging, AdaBoost, Random Forest and Stacking) have been considered.

Bagging (Bootstrap aggregating) algorithm [24] is a multiple classifier algorithm, which reduces the variance and avoids overfitting. Diversity among the base learners is achieved by bootstrap sampling from the original dataset to obtain training set. In bagging algorithm, replicated datasets are utilized to train base classification algorithms. The predictions of classification algorithms are combined by majority voting scheme. Bagging yields promising results when the base learners are unstable.

Boosting is a multiple classifier method, which aims to construct strong learners from weak learners (such as decision trees) by adjusting iteratively the weight of instances and the weights of classification methods. AdaBoost algorithm [25] is one of the most popular Boosting algorithm, with high predictive performance on several different application fields. AdaBoost algorithm is an adaptive algorithm, which aims to obtain a robust classification scheme by dedicating more iterations to harder instances [25].

Random Forests algorithm (RF) is an ensemble classification scheme, which combines tree predictors such that each tree grows in randomly selected subspaces of data [24]. In this scheme, a random feature selection is employed to split each node. In Random Forests, each tree in the ensemble is grown based on a random parameter and the final prediction of ensemble is obtained by aggregation. The predictive performance has been enhanced by growing an ensemble of trees and aggregating the trees by voting for the most popular class. Random Forests achieves high diversity by employing bootstrap aggregation (i.e. simple random sampling with replacement) and node splitting from a subset of total feature set.

Stacking (namely, stacked generalization) is another multiple classifier algorithm [26]. The other multiple classifier algorithms (such as Bagging and AdaBoost) are based on the combination of the same type weak learning algorithms. In contrast, Stacking algorithm obtains multiple classifier system by following a two-staged procedure. In the first level, the base learning algorithms are trained on the instances to estimate the class labels. In the second level, a combiner algorithm is trained on meta-instances. The meta-instances consists of the predictions of all algorithms.

3. PARTICLE SWARM OPTIMIZATION BASED STACKING

Particle swarm optimization (PSO) algorithm is a stochastic population-based algorithm, which simulates the social behaviour of organisms, such as bird flocking and fish schooling [27]. In PSO algorithm, each single candidate solution is denoted as a particle in the search space. Each particle has its own position (which corresponds to its direction) and its own velocity (which corresponds to its current direction). In this way, search space of possible solutions is explored by the particles. In PSO algorithm, search characteristics of particles are influenced by the cooperation among the particles of swarm. The algorithm has a simple structure and involves a small number of parameters [30].

There are many optimization problems with discrete-valued search spaces. To operate on binary search spaces, a discrete variant of PSO was developed [27]. In the binary PSO, particles correspond to the positions in the binary search space. The position vector of a particle can take the values of zero or one and the change of the position of a particle indicates flipping number of bits from one value to another. In this way, a particle can move on a hypercube by flipping the number of bits. In binary PSO, velocity values are limited to a range of [0, 1]. Hence, velocities and particle directions are regarded as the probability of finding the particle in one state or the other. To limit the range of velocity to [0, 1], a normalization method, such as sigmoid function can be utilized [28].

As emphasized by the earlier works mentioned in advance, an optimal configuration for Stacking algorithm can be modelled as an optimization problem [13-15]. In this work, binary particle swarm optimization algorithm is utilized in the construction of an appropriate configuration for Stacking ensemble [30]. Particle swarm optimization based optimization is conducted at two different levels: optimization at the base-level classifier selection and optimization at the base-level and meta-level classifier selection. These methods are referred as PSO-Stacking1 and PSO-Stacking2, respectively. The general framework for particle swarm optimization based Stacking is adapted from [14]. In this framework, particle swarm optimization algorithm is applied to search Stacking configurations, Stacking is trained and validation with training sets and validation sets, respectively and the best particle is obtained as a final configuration of Stacking. Then, this final configuration is evaluated by the testing set. In PSO-Stacking1, there is a pool of classifiers consisting of ten classification algorithms from which the optimal subset of classifiers will be selected. These classifiers include logistic regression, Naïve Bayes, C4.5, K-nearest neighbour, K-star, ZeroR, Decision Stump and PART algorithms. In this scheme, the meta-level classification algorithm is kept fixed and logistic regression method is utilized as the meta-level classifier. In contrast, PSO-Stacking2 has the same set of classifiers for base-level classification, whereas the meta-level classification algorithm is not kept fixed. Instead, ten classification algorithms examined for base-level classification are possible candidates of meta-level classification. The only difference between PSO-Stacking1 and PSO-Stacking2 is in the selection of meta-level classifier. Hence, we denote the general principles of particle swarm optimization based Stacking by PSO-Stacking. In PSO-Stacking, a binary string is employed to represent the position of each particle. The value of zero implies that the corresponding classifier is not selected, whereas the value of one implies that the corresponding classifier is selected. To evaluate the particles, F-measure is utilized as the fitness function. F-measure is the harmonic mean of precision and recall. Precision (PRE) is computed as given by Equation 1. Recall (REC) is computed as given by Equation 2. Based on the precision and recall values, F-measure is computed as given by Equation 3.

$$PRE = \frac{TP}{TP + FP} \tag{1}$$

$$REC = \frac{TP}{TP + FN} \tag{2}$$

$$F - measure = \frac{2 * PRE * REC}{PRE + REC} \tag{3}$$

Each particle is updated based on the equations as given by Equations 4-7, where $pbest_p$ denotes the best fitness value for each particle, *gbest* denotes the best fitness value within a group of $pbest_p$, w denotes the inertia weight, c_1 and c_2 denote acceleration parameters, rand, $rand_1$ and $rand_2$ are random numbers, v_{pd}^{old} and v_{pd}^{new} are velocities of old and new particles, respectively [30]:

$$v_{pd}^{new} = wv_{pd}^{old} + c_1 rand_1 (pbest_{pd} - x_{pd}^{old}) + c_2 rand_2 (gbest_d - x_{pd}^{old}) \tag{4}$$

$$\text{If } v_{pd}^{new} \notin (V_{\min}, V_{\max}) \text{ then } v_{pd}^{new} = \max(\min(V_{\max}, v_{pd}^{new}), V_{\min}) \tag{5}$$

$$S(v_{pd}^{new}) = \frac{1}{1 + e^{-v_{pd}^{new}}} \tag{6}$$

$$\text{If } (rand < S(v_{pd}^{new})) \text{ then } x_{pd}^{new} = 1 \text{ else } x_{pd}^{new} = 0 \tag{7}$$

Randomly initialize particle swarm
While maximum number of iterations have not been reached
    Evaluate fitness value of particle swarm by the ensemble classification scheme.
    For p=1 to number of particles
      If fitness of X_p is greated than the fitness of $pbest_p$ then
          $Pbest_p$=X_p
      endif
      If fitness of *gbest* is the same Max times then reset *gbest*
      Endif
      For d=1 to number of dimensions of particle

$$v_{pd}^{new} = wv_{pd}^{old} + c_1 rand_1 (pbest_{pd} - x_{pd}^{old}) + c_2 rand_2 (gbest_d - x_{pd}^{old})$$

If $v_{pd}^{new} \notin (V_{\min}, V_{\max})$ then $v_{pd}^{new} = \max(\min(V_{\max}, v_{pd}^{new}), V_{\min})$

$$S(v_{pd}^{new}) = \frac{1}{1+e^{-v_{pd}^{new}}}$$

If $(rand < S(v_{pd}^{new}))$ then $x_{pd}^{new} = 1$ else $x_{pd}^{new} = 0$

Next *d*

Next *p*

Next generation until stopping criterion is met

Figure 1. The general structure for PSO-Stacking1

The general structure of PSO-Stacking1 is outlined in Figure 1, where the maximum number of iterations is set to 100, rand, $rand_1$ and $rand_2$ are random numbers in the range of [0, 1], c_1 and c_2 acceleration parameters are set to 2 based on the earlier empirical results [27, 30]. As emphasized in advance in PSO-Stacking1, meta-level classification algorithm remains fixed during the evaluation of the ensemble configuration. In PSO-Stacking2, on the other hand, any classification algorithm can perform the meta-learning task. Hence, the framework of PSO-Stacking1 is modified as follows: For each base-level classification algorithm configuration, fitness values with different meta-learners (logistic regression, Naïve Bayes, C4.5, K-nearest neighbour, K-star, ZeroR, Decision Stump and PART algorithms) are examined and the configuration with the highest F-measure value is selected.

4. RESULTS AND DISCUSSIONS

In this section, datasets, evaluation measure, empirical settings and experimental results are given.

4.1. Text Collections

To empirically analyze the classification accuracy of PSO-Stacking approaches, we have employed four text categorization datasets from web pages domain. The descriptive information regarding the datasets is summarized in Table 1, where text collections are represented via unigram data representation scheme. In the experimental evaluations, we have utilized latent Dirichlet allocation (LDA) based representation with Gibbs sampling for text documents [31-32].

Table 1. Descriptive information for text collections [12]

Text Collection	Number of documents	Number of features	Number of classes
DMOZ-Business-500	18500	8303	37
DMOZ-Computers-500	9500	5011	19
DMOZ-Science-500	6000	4821	12
DMOZ-Sports-500	13500	5682	27

4.2. Evaluation Metric

In the empirical evaluation, classification accuracy (ACC) is employed as the evaluation criteria. Classification accuracy is computed as given by Equation 8:

$$ACC = \frac{TN+TP}{TP+FP+FN+TN} \quad (8)$$

where *TN*, *TP*, *FP* and *FN* represents true negatives, true positives, false positives and false negatives, respectively.

4.3. Empirical Settings

In the empirical analysis, k-fold cross validation method (k=10) is utilized. The result reports the average performance across all ten trials. The experiments are done on WEKA 3.7.11. The metaheuristic based ensemble configuration schemes are also implemented in Java. For each algorithm, the default set of parameters of WEKA are considered. To obtain an optimal configuration for Stacking ensemble, ten classification algorithms in WEKA are taken into account. In PSO-Stacking1, these classifiers are candidates for base-level classification. In PSO-Stacking2, these classifiers are candidates for base-level and meta-level classification. In order to be consistent with the earlier work on the use of metaheuristics to determine the optimal configuration for Stacking, ten classifiers are the same with these references. Hence, logistic regression, Naïve Bayes, C4.5, K-nearest neighbour, K-star, ZeroR, Decision Stump and PART algorithms are selected [14, 15].

In this classifier pool, there are classification algorithms from various classification approaches. Hence, the diversity involved in the classifier ensemble construction is achieved. In the empirical analysis, the stacking configuration found by PSO-stacking is examined with several base learners and ensemble approaches. First, it is compared to base-classifiers, such as logistic regression, Naïve Bayes, C4.5, K-nearest neighbour, K-star, ZeroR, Decision Stump and PART algorithms. Secondly, it is compared to conventional ensemble approaches, such as Bagging, AdaBosot, Random

Forest, Stacking and StackingC. Bagging with RepTree decision tree as its base-level learning algorithm, Bagging with C4.5 decision tree as its first-level learning algorithm, AdaBoost with Decision Stump as its base-level learning algorithm, AdaBoost with C4.5 decision tree as its first-level learning algorithm are examined.

Naïve Bayes, C4.5 and K-nearest neighbour algorithms are utilized in Stacking and StackingC based ensemble construction. Logistic regression is employed as a meta-learner in Stacking and linear regression is employed as a meta-learner in StackingC. For PSO-Stacking1, logistic regression method is utilized as the meta-level classifier. Besides, PSO-Stacking approaches are compared to the other metaheuristic based configuration optimization approaches, such as GA-Stacking, ACO-Stacking and ABC-Stacking. For these algorithms, the parameters assigned the values mentioned in the corresponding references [14, 15].

4.4. Experimental Results

In Table 2, the average classification accuracy of classification algorithms and multiple classifier methods have been presented, where the best results for a particular configuration are denoted by using boldface. First, we examined the classification accuracy enhancement of PSO-stacking over individual base learning algorithms. The particle swarm optimization based configurations of Stacking yield better classification accuracy than individual learning algorithms. Regarding results of learning algorithms, the highest results are achieved by logistic regression and Naïve Bayes classifiers.

Secondly, we examined the classification accuracy of PSO-Stacking and the other well-known multiple classifiers, such as Bagging, AdaBoost, Random Forest, Stacking and StackingC. Particle swarm optimization based Stacking ensemble configuration generally yields better classification accuracy than the other well-known multiple classifier methods. Besides, Stacking and StackingC multiple classifiers generally obtain better classification accuracies than multiple classifier methods, such as Bagging, AdaBoost and Random Forest.

Regarding the classification accuracys of PSO-stacking and other metaheuristic multiple classifiers, the highest classification accuracies among the metaheuristic based ensemble configurations for Stacking is obtained by PSO-Stacking2. The second highest classification accuracies for metaheuristic based ensemble configurations for Stacking are obtained by PSO-Stacking1, ACO-Stacking2, ABC-Stacking2 depending on the different datasets.

Table 2. Classification accuracies of compared algorithms on text categorization benchmarks

Algorithm	DMOZ-Business-500	DMOZ-Computers-500	DMOZ-Science-500	DMOZ-Sports-500
LR	47.06	54.25	56.37	64.63
NB	39.24	47.06	50.42	52.61
C4.5	32.97	39.73	40.30	54.66
KNN (k=1)	34.63	38.05	43.33	52.32
KNN (k=2)	38.65	42.01	47.37	52.49
K-star	37.82	42.07	40.08	46.49
OneR	39.75	43.25	38.05	39.10
PART	34.70	37.86	44.46	55.13
ZeroR	30.24	29.95	29.87	29.92
Decision Stump	35.40	38.04	29.91	38.10
Bagging (REP Tree)	34.11	39.72	42.89	56.97
Bagging (C4.5)	58.17	62.12	63.90	73.81
AdaBoost (Decision Stump)	56.80	62.60	44.35	57.62
AdaBoost (C4.5)	56.49	60.95	63.74	73.51
Random Forest	57.06	63	52.37	66.16
Stacking	59.75	63.46	64.81	73.66
StackingC	59.81	63.35	64.14	73.49
GA-Stacking1	61.2	65.23	66.11	74.76
GA-Stacking2	61.55	65.50	67.31	74.8
ACO-Stacking1	61.18	66.80	66.37	73.97
ACO-Stacking2	63	67.35	66.04	75.58
ABC-Stacking1	63.80	66.87	66.47	73.13
ABC-Stacking2	64.70	68.31	67.35	74.55
PSO-Stacking1	63.59	67.87	67.7	75.56

PSO-Stacking2	**66.18**	**70.01**	**72.12**	**79.52**

5. CONCLUSION

Multiple classifier system is a research field of pattern recognition, aiming to integrate individual classifiers to obtain a scheme with better classification performance. This study introduces a multiple classifier approach to text categorization. In this scheme, Stacking method is utilized to construct ensembles.

The two configuration issues of Stacking, i.e. base-level classifier selection and meta-level classifier selection are addressed by the use of binary particle swarm optimization. The proposed approach is empirically evaluated with base learners, well-known techniques (such as Bagging, AdaBoost, Random Forest, Stacking and StackingC) and other metaheuristic based Stacking methods (such as ACO-Stacking and ABC-Stacking) in terms of classification accuracies. The experimental results indicate that PSO-Stacking can outperform conventional classification methods for text categorization.

REFERENCES

[1] L.I. Kuncheva, Combining pattern classifiers, methods and algorithms, New York: Wiley InterScience, 2005.
[2] J.Kittler and F. Roli, Multiple classifier systems, Berlin: Springer, 2000.
[3] L.Rokach, "Ensemble-based classifiers", Artificial Intelligence Review, vol.33, pp.1-39, 2010.
[4] F.J.Provost and V.Kolluri, "A survey of methods for scaling up inductive learning algorithms", Data Mining and Knowledge Discovery, vol. 3, pp. 131-169, 1999.
[5] A.Onan, S.Korukoğlu and H.Bulut, "A multiobjective weighted voting ensemble classifier based on differential evolution algorithm for text sentiment classification", Expert Systems with Applications, vol. 62, pp. 1-16, 2016.
[6] D.H.Wolpert, "Stacked generalization", Neural Networks, vol. 5, no 2, pp.241-259, 1992.
[7] M.Sewell, "Multiple classifier" UCL, Department of Computer Science Technical Report RN-11-02, (2011).
[8] R.Polikar, "Ensemble based systems in decision making", IEEE Circuits and Systems Magazine, vol. 6, no 3, pp. 21-45, 2006.
[9] M.P.Sesmero, A.I.Ledezma and A.Sanchis, "Generating ensembles of heterogeneous classifier using stacked generalization", Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, vol. 5, no 1, pp. 21-34, 2015.
[10] E.G.Talbi, Metaheuristics from design to implementation, New York: Wiley, 2009.
[11] A. Gogna and A.Tayal, "Metaheuristics: review and application", Journal of Experimental & Theoretical Artificial Intelligence, vol. 25, no 4, pp. 503-526, 2013.
[12] A.Onan, H.Bulut and S.Korukoğlu, "An improved ant algorithm with LDA-based representation for text document clustering", Journal of Information Science, vol.43, no 2, pp.275-292, 2017.
[13] H.A.Abbas, C.S.Newton and R.Sarkar, Heuristic search-based stacking of classifiers, Berlin: Springer, 2002.
[14] Y.Chen and M.L.Wong, "An ant colony optimization approach for stacking ensemble", in Proceedings of the Second World Congress on Nature and Biologically Inspired Computing, (December 15-17, Kitakyushu, Japan, 2010), 146-151, (2010).
[15] P.Shunmugapriya and S.Kanmani, "Optimization of stacking ensemble configurations through artificial bee colony algorithm", Swarm and Evolutionary Computation, vol.12, pp.24-32, 2013.
[16] M.Kantardzic, Data mining: concepts, models, methods and algorithms, New York: Wiley-IEEE Press, 2011.
[17] G.Shmueli, N.R.Patel and P.C.Bruce, Data mining for business intelligence: concepts, techniques, and applications in Microsoft Office Excel with XLMiner, New Jersey: John Wiley & Sons, 2010.
[18] R.Quinlan, C4.5: programs for machine learning, San Mateo: Morgan Kaufmann, 1993.
[19] X.Niuniu and L.Yuxun, "Review of decision trees", in Proceedings of the third IEEE International Conference on Computer Science and Information Technology, (July, Chengdu, China, 2010), 105-109, (2010).
[20] D.W.Aha, D.Kibler and M.K.Albert, "Instance based learning algorithm", Machine Learning, vol.6, pp.37-66, 1991.
[21] J.G.Clearly and L.E.Trigg, "K*: an instance-based learner using an entropic distance measure", in Proceedings of the twelfth international conference on machine learning, (July 9-12, Tahoe City, California, 1995), 108-114, (1995).
[22] W.Iba and P.Langley, "Induction of one-level decision trees", in Proceedings of the 9th International Workshop on Machine Learning, (Aberdeen, UK, 1992), 233-240, (1992).
[23] E.Frank and I.H.Witten, "Generating accurate rule sets without global optimization", in Proceedings of the 15th International Conference on Machine Learning, (July 24-27, 1998), 144-151, (1998).
[24] L.Breiman, "Bagging predictors", Machine Learning, vol. 4, no 2, pp. 123-140, 1996.
[25] Y.Freund and R.E.Schapire, "Experiments with a new boosting algorithm", in Proceedings of the Thriteenth International Conference on Machine Learning, (July 3-6, 1996), 1-9, (1996).
[26] D.H. Wolpert, "Stacked generalization", Neural Networks, vol.5, no 2, pp. 241-259, 1992.
[27] J.Kennedy and R.C.Eberhart, "Particle swarm optimization", in Proceedings of the International Conference on Neural Networks, 1942-1948, (1995).
[28] A.P. Engelbrecht, Computational intelligence: an introduction, New York: Wiley, 2007.
[29] M.N.A.Wahab, S.Nefti-Meziani and A.Atyabi, "A comprehensive review of swarm optimization algorithms", Plos One, doi: 10.1371/journal.pone.0122827
[30] L.Y.Chuang, H.W.Chang, C.J.Tu and C.H.Yang, "Improved binary PSO for feature selection using gene

expression data", Computational Biology and Chemistry, vol. 32, pp.29-38, 2008.
[31] A.Onan, S.Korukoğlu and H.Bulut, "LDA-based topic modelling in text sentiment classification: an empirical analysis", International Journal of Computational Linguistics and Applications, vol.7,no 1, pp. 101-119, 2016.
[32] A.Onan, "Hybrid supervised clustering based ensemble scheme for text categorization", Kybernetes, vol. 46, no 2, pp. 330-348, 2017.