Neurocomputing 73 (2010) 3319–3322

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Letters

On the evolutionary design of heterogeneous Bagging models

André L.V. Coelho *, Diego S.C. Nascimento

University of Fortaleza, Graduate Program in Applied Informatics, Av. Washington Soares 1321, 60811-905 Fortaleza, Brazil

ARTICLE INFO

Article history:
Received 21 December 2009
Received in revised form
11 May 2010
Accepted 18 July 2010
Communicated by Y. Liu
Available online 26 August 2010

Keywords:
Ensembles
Bagging
Heterogeneous models
Genetic algorithms
Classification

ABSTRACT

Bagging is a popular ensemble algorithm based on the idea of data resampling. In this paper, aiming at increasing the incurred levels of ensemble diversity, we present an evolutionary approach for optimally designing Bagging models composed of heterogeneous components. To assess its potentials, experiments with well-known learning algorithms and classification datasets are discussed whereby the accuracy, generalization and diversity levels achieved with heterogeneous Bagging are matched against those delivered by standard Bagging with homogeneous components.

1. Introduction

Over the last decades, the strategy of combining multiple classifiers into ensembles has received increasing interest due to its potential in bringing about significant improvements in terms of training accuracy and learning generalization [1,2]. As the key for the success of any ensemble lies in how its components disagree on their predictions [3], several approaches for designing diverse components have been conceived, among which those using different subsets of training data jointly with a single learning method [4,5] and those adopting different learning methods associated with different predictors [6,7].

A well-known representative of the first group is Bagging, which is based on the idea of data resampling [4,5,8,9]. Diversity is promoted in Bagging by using bootstrapped replicas of the training dataset, each replica being generated by randomly drawing, with replacement, a subset of the training data. Typically, each new dataset will have the same number of instances of the original dataset; however, since some instances will appear repeatedly while others will not show up, the effective size will be lower and the datasets will overlap significantly. Each derived dataset is used to train a classifier, and then, for any test instance, the outputs of the individual classifiers are aggregated via the simple majority vote (MV) rule. Usually, unstable classifiers are adopted as base models, since this type of classifier can generate sufficiently different decision boundaries even for small perturbations in the training parameters [2,4].

In this paper, aiming at further increasing the diversity levels of the ensemble models produced by Bagging, we present an evolutionary approach for optimally designing Bagging models composed of heterogeneous components. Even though the idea of heterogeneous ensembles has been recently advocated [6,7], so far there is no deep investigation on the benefits of adopting different learning algorithms in the context of Bagging. In fact, this idea seems very reasonable since different classes of learning algorithms are usually associated with different search/representation biases (and thus hypothesis spaces) [10], thereby fomenting ensemble diversity. Since the automatic configuration of the best heterogeneous Bagging (HB) model for a given classification problem turns out to be a combinatorial optimization problem in itself, a customized genetic algorithm (GA) [11] has been adopted for this purpose. To validate the novel approach, experiments with well-known learning algorithms and classification datasets are reported here whereby the accuracy, generalization and diversity levels incurred with HB are matched against those delivered by standard Bagging with homogeneous components.

In the sequel, we describe how HB models are evolved and then discuss the results achieved in the experiments. The last section concludes the paper, bringing remarks on future work.

2. Evolving heterogeneous Bagging models

In a nutshell, HB allows that different learning algorithms be recruited to induce the ensemble components over the resampled

---

* Principal corresponding author. Tel.: +55 85 34773268; fax: +55 85 34773061.
E-mail addresses: acoelho@unifor.br, coelho.alv@gmail.com (A.L.V. Coelho), silveiraal@gmail.com (D.S.C. Nascimento).

0925-2312/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2010.07.008

data produced via Bagging. For this purpose, we have resorted to $M=10$ state-of-the-art learning algorithms representing five distinct classes of classifier inducers [10]: (i) simple Naïve Bayes (NB), founded on Bayesian statistics; (ii) RBF neural networks (NN) and support vector machines (SVM) trained with SMO algorithm, both based on nonlinear function representations; (iii) J48, decision stump (DS), and REP tree (RT), working with decision trees; (iv) IBk, an instance learning algorithm; and v) OneR, PART and decision table (DT), which generate hypothesis in rule format. This repertory covers both simple and complex inducers; likewise, it includes unstable (NN, J48, PART) and stable (IBk, NB, SVM) classifiers. By this means, it is possible to assess whether the fusion of simple with complex, unstable with stable classifiers via HB could bring about gains in accuracy and generalization.

One should note that the application of distinct HB structures may entail very different results in terms of ensemble performance. As the size of the space of possible heterogeneous structures is exponential in nature, a customized GA [11] has been deployed for optimally designing HB templates. Each individual of the GA population P encodes a whole HB structure and is codified as a K-size string of integer values, where K is the maximum number of components allowed. For filling in the k-th position (gene), $M+1$ values (alleles) are available, one for each type of inducer in the above-mentioned repertory and another indicating the possibility of component pruning. Usually, the pruning of ensemble components happens as a second stage in ensemble creation (after the generation of components) in order to reduce redundancy and complexity of the resulting ensemble model [12]. In our case, component generation and pruning occur simultaneously, allowing HB to tune the ensemble size in agreement with the problem's demands.

To evaluate the quality of HB templates, a convex linear combination of two terms has been adopted as fitness function: one related to accuracy and the other to parsimony. While the first refers to the cross-validation error delivered in training (see next section), the latter measures the complexity of an HB model. By this means, the lower the cross-validation error and the number of components of an ensemble model are, the higher will be the associated fitness value. Moreover, Roulette Wheel [11] is used both for selecting individuals to reproduce (among parents) and to survive to the next generation (among parents and offspring), even though elitism [11] is also adopted in the replacement phase. Individuals are recombined through a single-point crossover and the resulting offspring undergo modifications via random-resetting mutation [11]. The stop criterion adopted has been to go through a given number of generations of evolution.

3. Empirical assessment

To assess the potentials of HB, a prototype was developed under Weka [10] and extensive experiments were conducted over 20 UCI benchmark datasets [13]. These datasets (two of which contain noisy data, viz. *monks-3* and *waveform*) are indicated in Tables 1 and 2, and their description in terms of type/number of attributes and number of instances/classes can be found elsewhere [5,9]. To serve as yardstick against which we could match the performance of HB models, we have also recorded results delivered by standard, homogeneous Bagging induced with each of the M types of learning algorithms described earlier.

Aiming at producing statistically significant results, for each dataset, 10 pairs of stratified training/test (66.6%/33.4%) partitions were randomly generated by using different random seeds. Through a 10-fold stratified cross-validation [10] over the training partitions, both homogeneous and heterogeneous Bagging models were induced and assessed. The resulting error rates served as fitness scores to guide the GA engine. Conversely, test data were used for assessing the levels of generalization achieved by the resulting ensemble models trained ultimately over the whole training partition.

For the experiments, the GA control parameters were set as follows (after manual calibration): 20 as population size (n_p); 80% and 10% as crossover and mutation rates, respectively; and 20 as maximum number of generations (n_g). It is worth mentioning that the performance results reported for each dataset relate to the best weight combinations achieved for the two terms employed in the GA fitness function. Besides, we have made extensive use of the validation testbench and code implementations (with default control parameter values) available in Weka [10] for the M learning algorithms adopted in HB (see Section 2). Moreover, although we have experimented with different values for the

Table 1
Performance comparison between homogeneous and heterogeneous Bagging models—accuracy.

Dataset	**Homogeneous Bagging**		**Heterogeneous Bagging**		
	Best (type)	Average	Best	Size	Comp. types
anneal	0.0252 ± 0.0082 (DT)	0.1228 ± 0.0481	**0.0251** ± **0.0059**	9.1 ± 1.1	DT,NB,SVM
breast-cancer	0.2673 ± 0.0446 (DS)	0.2876 ± 0.0169	**0.2224** ± **0.0333**	9.4 ± 0.7	NB,NN,OneR
bupa	0.3280 ± 0.0438 (PART)	0.4009 ± 0.0411	**0.3102** ± **0.0302**	8.9 ± 0.3	RT,DS,PART
colic	0.1548 ± 0.0231 (PART)	0.1916 ± 0.0236	**0.1206** ± **0.0284**	9.5 ± 0.5	J48,RT,DT
credit-a	0.1323 ± 0.0255 (PART)	0.1533 ± 0.0281	**0.1128** ± **0.0156**	8.9 ± 0.3	OneR,DT,NB
diabetes	0.2546 ± 0.0256 (NB)	0.2744 ± 0.0204	**0.2237** ± **0.0270**	8.9 ± 0.3	NB,SVM,PART
glass	0.3699 ± 0.0483 (J48)	0.4349 ± 0.0630	**0.3000** ± **0.0326**	8.9 ± 0.3	NN,J48,IBk
haberman	0.2600 ± 0.0110 (NB)	0.2835 ± 0.0225	**0.2333** ± **0.0197**	9.0 ± 0.0	DS,OneR,NN
heart-c	0.1606 ± 0.0317 (NN)	0.2146 ± 0.0352	**0.1413** ± **0.0231**	8.9 ± 0.3	NB,NN,RT
hepatitis	0.1358 ± 0.0406 (NB)	0.1943 ± 0.0270	**0.1132** ± **0.0487**	8.7 ± 0.5	NB,PART,DS
ionosphere	0.1067 ± 0.0207 (NN)	0.1417 ± 0.0330	**0.0650** ± **0.0146**	9.0 ± 0.0	NB,NN,OneR
iris	0.0451 ± 0.0132 (NN)	0.0714 ± 0.0256	**0.0294** ± **0.0139**	8.9 ± 1.2	NN,IBk,RT
monks-3	0.0653 ± 0.0534 (SVM)	0.1398 ± 0.5660	**0.0414** ± **0.0346**	8.7 ± 0.7	PART,SVM,RT
segment	0.0525 ± 0.0087 (PART)	0.1492 ± 0.1423	**0.0380** ± **0.0034**	9.0 ± 0.0	DT,PART,J48
sick	0.0175 ± 0.0040 (J48)	0.0390 ± 0.0214	**0.0157** ± **0.0031**	9.4 ± 0.7	J48,PART,RT
sonar	0.2141 ± 0.0643 (IBk)	0.2968 ± 0.0392	**0.1746** ± **0.0416**	8.8 ± 0.4	IBk,NN,SVM
vehicle	0.2806 ± 0.0189 (PART)	0.3783 ± 0.1154	**0.2497** ± **0.0170**	9.1 ± 0.9	PART,RT,SVM
vote	0.0466 ± 0.0137 (RT)	0.0586 ± 0.0169	**0.0385** ± **0.0115**	8.9 ± 0.3	RT,PART,IBk
waveform	**0.1443** ± **0.0057** (SVM)	0.2349 ± 0.1041	0.1444 ± 0.0072	9.1 ± 1.0	SVM,NN,PART
zoo	**0.0857** ± **0.0269** (IBk)	0.2717 ± 0.1933	0.0886 ± 0.0250	8.9 ± 0.3	SVM,IBk,NN