Empirical evaluation of ensemble techniques for a Pittsburgh Learning Classifier System

Jaume Bacardit
ASAP research group, School of Computer Science and IT, University of Nottingham, Jubilee Campus, Nottingham, NG8 1BB, UK
jqb@cs.nott.ac.uk

Natalio Krasnogor
ASAP research group, School of Computer Science and IT, University of Nottingham, Jubilee Campus, Nottingham, NG8 1BB, UK
nxk@cs.nott.ac.uk

ABSTRACT

Ensemble techniques have proved to be very useful to boost the performance of several types of machine learning methods. In this paper, we illustrate its usefulness in combination with GAssist, a Pittsburgh-style Learning Classifier System. Two types of ensemble are tested. First bagging-style consensus prediction. Second an ensemble intended to deal more efficiently with ordinal classification problems. Both methods improve the performance and behaviour of GAssist in the tested domains.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning—*Concept Learning, Induction*

General Terms

Algorithms,Experimentation,Performance

Keywords

Evolutionary Algorithms, Learning Classifier Systems, Rule Induction, Ensemble techniques

1 Introduction

Ensemble learning, a family of techniques established for more than a decade in the Machine Learning community provides performance boost and robustness to the learning process by integrating the collective predictions of a set of models in some principled fashion [13]. This family of techniques covers many different approaches, the two most representative methods being Bagging [5] and Boosting [8].

This paper presents the empirical evaluation of two types of ensemble techniques that use Learning Classifier Systems (LCS) to generate the models that will be integrated. Specifically we will use a recent LCS called GAssist [1] that belongs to the Pittsburgh approach of LCS.

The first of these two approaches will consist in a simple consensus voting of an ensemble of rule sets generated by running GAssist several times on the same dataset with different initial random seeds. This is conceptually similar to Bagging, but its mechanics are even simpler.

The second type of ensemble is designed to solve problems of ordinal classification [7]. That is, when the classes of the problem have some intrinsic order. This approach divides an N classes dataset into N-1 binary datasets structured in a hierarchical way. The ensemble duty is to integrate the predictions of these N-1 binary models into a final prediction of the N classes domain.

The rest of the paper is structured as follows: First, section 2 will describe some related work. Next, section 3 will contain the main characteristics of GAssist, the Pittsburgh LCS used in this paper. Section 4 will contain the description and empirical evaluation of the first type of ensemble studied in this paper, the consensus voting ensemble, and section 5 of the second type, the hierarchical ensemble for ordinal classification. Finally, section 6 will describe the conclusions and further work of the paper.

2 Related work

Usually, there are two questions that have to be addressed when building and using an ensemble that integrates the predictions of several models:

- What data is used to train each model?
- How are the individual model predictions integrated to produce a final ensemble prediction?

In the case of Bagging [5], N views of the training set are generated by a sampling with replacement procedure, and each of the N models in the ensemble is trained with one of these views. After that, the ensemble prediction process follows a simple majority voting: the ensemble will predict the most frequent class from the ones predicted by the N members of the ensemble.

The aim of bagging is to generate models that complement each other. This is achieved implicitly by the sampling process used to generate the models. On the other hand, Boosting [8] achieves the same aim in an explicit way. This method generates the models and the dataset in an iterative way. The first of them uses the original training data, and the later models focus on learning the examples that were mis-classified by the previous models. This is achieved by weighting the instances based on their mis-classification rate on previous models. Finally, the ensemble prediction is a weighted voting process, where the weight of a model is based on its error over the training data used to generate it.

There are few examples of the use of ensembles in the LCS community. Llorà et al. [11] studied several policies to select the representative candidates from the final population of a Pittsburgh LCS, but did not integrate individuals from several populations. Also, Bull et al. [9] investigated the use of an ensemble of whole LCS populations using an island model. Their ensemble took place during the learning process, not afterwards unlike the approach of this paper.

There are several methods reported in the literature to perform ordinal classification by means of an ensemble. For instance, the method proposed by Frank and Hall [7] takes advantage of learning techniques that can produce a probability of an instances belonging to a certain class and divides the learning process of an N class ordinal domain into N-1 binary domains in the following way:

1. This method needs models that can produce class probability estimates
2. For a given domain D with ordered classes ranging from 1 to k
3. $k-1$ binary domains D_i .. D_{k-1} are generated, where the class definition for domain i will be defined by the predicate $D > i$. That is, the subdomain D_1 will predict if the class of the examples is greater than 1, D_2 will predict if the class of the examples is greater than 2, ...
4. Models for these $k-1$ domains are generated
5. For each new unseen instances, the probability that this instance belongs to each of the k classes is computed as follows:
   - $P(D = 1) = 1 - P(D > 1)$
   - $P(D = i) = P(D > i - 1) - P(D > i), 1 < i < k$
   - $P(D = k) = P(D > k - 1)$
6. The ensemble predicts the class with higher probability

3 The GAssist Learning Classifier System

GAssist [1] is a Pittsburgh Genetic–Based Machine Learning system descendant of GABIL [6]. The system applies a near-standard generational GA that evolves individuals that represent complete problem solutions. An individual consists of an ordered, variable–length rule set. Details of all the components of the system and standard parameters are described in [2].

4 Ensembles for consensus prediction

The evaluated ensemble technique follows these steps:

1. GAssist is run N times on the unmodified training set, each time using a different seed to initialize the pseudo-random numbers generator
2. From each of these N runs a rule set is extracted. This rule set corresponds to the best individual of the population, evaluated using the training set
3. For each instance in the test set, the N members of the ensemble produce a prediction. The majority class of these predictions is used

This ensemble technique is very similar to Bagging, with just one difference: all models (rule sets) are learned using the same training data: the original training set.

Each rule set produced by GAssist is stored in text format as its phenotype representation: the actual ordered predicates in conjunctive normal form that constitute a rule set. Moreover, when these rule sets are dumped to text format, only the relevant attributes are expressed. This means that the ensemble code will not make unnecessary calculations for the match process of the irrelevant attributes. To illustrate the text format used to express the rules generated by GAssist, figure 1 contains an example of a rule set for the Wisconsin Breast Cancer domain generated by GAssist.

4.1 Empirical evaluation

In order to evaluate the performance of the ensemble method described in this paper we have used a set of 25 datasets that represent a broad range of domains in respect to number of attributes, instances, type, etc. These problems were taken from the University of California at Irvine (UCI) repository [4]. The characteristics of the datasets are summarized in [2]. The datasets are partitioned using the standard stratified ten-fold cross-validation method. Three different sets of 10-cv folds have been used. Also, the experiments were repeated 15 times with different random seeds. This means that the GAssist results for each dataset included 450 runs, either by averaging the test accuracy of each of these 450 runs or by using the ensemble technique.

Student t-tests with a confidence interval of 95% were used to determine wether significant differences between the performance of the individual runs of GAssist and the ensemble of these same runs can be measured. The input data for the t-test will be the test accuracy obtained in each of the 30 test sets that we have (3x10-cv). The ensemble code produced one accuracy measure for each test set. The test accuracy of the individual runs of GAssist (15 repetitions for each data set) were computed by averaging these accuracies. The parameters of the system are the ones defined in [2].

Table 1 contains the results of the experiments performed to evaluate the ensemble technique studied in this paper. The table contains, for each dataset, the average accuracy of the 450 individual runs and the average accuracy of the 30 ensembles produced from the individual runs. We can observe how, for all datasets, the ensemble produces higher accuracy than the individual GAssist runs. The average accuracy increase is 2.5%. Also, the accuracy difference was significant in 10 of the 25 datasets, according to the t-tests.

5 Ensembles for ordinal classification

Our ensemble-based approach at ordinal classification is divided in two parts: (1) decomposition of the original N classes dataset into several binary sub-datasets and (2) integration of the models generated for each dataset to produce a final N-classes prediction.

The generation of the binary datasets works as follows:

1. We have an original dataset with N ordinal classes
2. We select a certain cut-point between the classes with the following criterion: we will select the cut-point that produces the most balanced sets in terms of number of instances at left and right of the cut-point
3. We transform the N classes dataset into a binary dataset: instances belonging to classes below the cut-

Figure 1: Rule set generated by GAssist for the Wisconsin Breast Cancer dataset

1:Att Clump_Thickness is [<9.4] and Att Cell_Size_Uniformity is [<4.6] and Att Cell_Shape_Uniformity is [<6.4] and Att Marginal_Adhesion is [<7.75] and Att Single_Epi_Cell_Size is [<5.5][>7.75] and Att Bare_Nuclei is [<5.5] and Att Normal_Nucleoli is [<4][5.5,8.5] and Att Mitoses is [<6.76][>7.12] → benign

2:Att Cell_Size_Uniformity is [<1.9] and Att Single_Epi_Cell_Size is [<7.75] and Att Normal_Nucleoli is [<4][5.5,8.5] → benign

3:Default rule → malignant

Table 1: Results of the experiments to evaluate the consensus ensemble applied over GAssist runs. A • symbol in a row means that the ensemble was able to significantly outperform the GAssist individual runs according to the t-tests

Dataset	GAssist acc.	Ensemble acc.
bal	79.0±4.0	82.5±3.8•
bpa	62.4±7.8	65.7±7.7
bre	70.5±7.9	73.0±7.6
cmc	54.4±3.9	55.7±3.6
col	93.3±4.3	96.2±3.2•
cr-a	85.1±4.1	86.0±3.7
gls	66.8±9.5	71.9±8.0•
h-c1	80.4±5.9	83.0±5.2•
h-h	95.7±3.4	96.7±2.7
h-s	80.2±7.6	82.0±6.9
hep	89.8±8.0	93.6±5.5•
ion	92.0±5.2	93.1±5.1
irs	95.3±5.6	95.8±5.6
lab	98.1±5.4	100.0±0.0•
lym	80.8±11.2	84.4±9.9
pim	74.7±4.8	75.6±4.0
prt	47.5±6.7	52.7±6.8•
son	76.6±9.3	84.0±7.5•
thy	92.0±5.7	93.8±5.4
vot	97.1±3.3	97.6±2.7
wbcd	96.1±2.5	96.2±2.3
wdbc	94.3±3.1	95.2±2.7
wine	93.4±5.5	96.3±3.9•
wpbc	75.3±8.3	80.4±7.6•
zoo	92.1±8.0	94.1±6.3
ave	82.5±13.7	85.0±12.9

point will be labelled as class 0. Instances belonging to classes over the cut-point will be labelled as class 1

4. The steps 2 and 3 will be repeated recursively for the instances to the left and to the right of the cut-point, until we arrive to the trivial case: having a binary dataset

This process will effectively convert a N classes ordinal classification domain into N-1 binary classification domains organized in a hierarchical way. The structure of this hierarchical ensemble is represented in figure 2. Unlike the Frank and Hall approach [7], these N-1 datasets will not have the same number instances as the original training set: The first dataset, corresponding to the root of the hierarchy will have the same number of examples as the original training set. Its two immediate descendants will have the number of examples at the left and right of the cut point, etc.

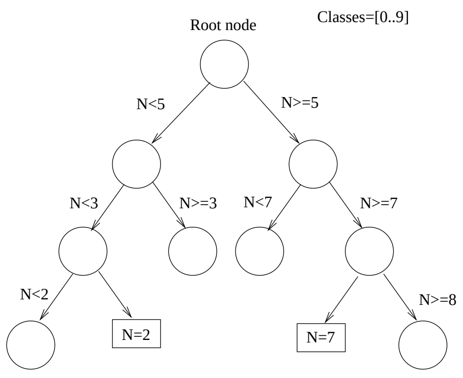
The ensemble predictions will be performed as follows:

1. For each test instance, we will query the model at the root of the hierarchy to determine if its class is lower or higher than the first cut-point.

2. If the root model predicts class 0, the next step is to predict this instance using the model generated for the left branch of the root node

3. If the root model predicts class 1, the next step is to predict this instance using the model generated for the right branch of the root node

Figure 2: Representation of the hierarchical ensemble for ordinal classification. Nodes (circles) with no descendants are already binary problems



4. The process will continue until we reach a leaf node

This process will also be faster than the Frank and Hall method, as only $log_2(k)$ models will be queried, instead of all $k-1$ of them. Moreover, the model used at each node to predict if a given instances is lower or higher than the cut point is an ensemble itself, using the method described in the previous sections.

5.1 Empirical evaluation of the hierarchical ensemble

We will use only one domain to illustrate the performance of this hierarchical ensemble. This domain belongs to the bioinformatics field, and specifically to protein structure prediction (PSP) [3]. Proteins are heterogeneous molecules that have a complex 3D structure very difficult to determine experimentally and therefore needs to be predicted. Several different features can be predicted from a proteins 3D structure. The domain used in this paper is called prediction of average solvent accessibility [12]. This property is real-valued and therefore we need some criterion to convert it into an ordinal set of classes. We will use the equal frequency discretization algorithm [10] for this task, dividing the domain into 10 ordinal classes.

The hierarchical ensemble obtained an accuracy of 23.9±3.0, while a flat ensemble of GAssist models performing normal classification, without any specific knowledge of the intrinsic class order, obtained an accuracy of 22.1±4.6. For reference, please note that Solvent Accessibility prediction accuracy for 10 classes with the kind of input information used in this paper ranges from 20 to 24% [12].

The performance difference is not significant, but the behaviours of both approaches is quite different: Table 2 contains the confusion matrix on one of the test folds for the hierarchical classifier, while table 3 contains the confusion matrix for the flat ensemble. The predictions of this domain are usually fed back into another PSP prediction task. Therefore it is important that, in the case of a mis-classification, the wrong predicted class is close to the real class in the intrinsic class order. The hierarchical classifier achieves this objective much better that the flat ensemble, as most of the predictions appear quite close to the diagonal. Numerically, we can compute the behaviour difference as the average misclassification penalty (AMP), defining the misclassification penalty as the distance in the intrinsic class order between the real and predicted classes of each test instance. The hierarchical ensemble obtained an AMP of 1.7±0.2, while the flat ensemble obtained an AMP of 2.0±0.2. In this case, the AMP difference between both systems was significant, according to the t-tests with 95% confidence level.

Table 2: Confusion matrix of the hierarchical classifier on a test set of the average solvent accessibility domain. Each cell contains the percentage of instances of the class in the row predicted as the class in the column

		Predicted class									
		0	1	2	3	4	5	6	7	8	9
Real class	0	50.0	0.0	0.0	33.3	16.7	0.0	0.0	0.0	0.0	0.0
	1	0.0	11.1	0.0	55.6	22.2	0.0	0.0	0.0	11.1	0.0
	2	18.2	0.0	9.1	36.4	18.2	0.0	18.2	0.0	0.0	0.0
	3	0.0	10.0	10.0	30.0	30.0	0.0	0.0	10.0	10.0	0.0
	4	7.7	0.0	0.0	7.7	30.8	23.1	15.4	15.4	0.0	0.0
	5	0.0	0.0	0.0	0.0	25.0	0.0	50.0	25.0	0.0	0.0
	6	0.0	0.0	0.0	25.0	0.0	0.0	37.5	25.0	12.5	0.0
	7	0.0	0.0	0.0	0.0	0.0	11.1	22.2	22.2	44.4	0.0
	8	0.0	0.0	0.0	0.0	0.0	5.6	11.1	44.4	27.8	11.1
	9	0.0	0.0	0.0	0.0	0.0	8.3	0.0	33.3	25.0	33.3

Table 3: Confusion matrix of the flat ensemble on a test set of the average solvent accessibility domain. Each cell contains the percentage of instances of the class in the row predicted as the class in the column

		Predicted class									
		0	1	2	3	4	5	6	7	8	9
Real class	0	83.3	0.0	0.0	0.0	0.0	16.7	0.0	0.0	0.0	0.0
	1	77.8	11.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.1
	2	54.5	18.2	0.0	0.0	0.0	27.3	0.0	0.0	0.0	0.0
	3	50.0	30.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	20.0
	4	30.8	7.7	0.0	0.0	0.0	30.8	7.7	0.0	15.4	7.7
	5	25.0	0.0	0.0	0.0	0.0	25.0	0.0	0.0	50.0	0.0
	6	12.5	12.5	0.0	0.0	0.0	12.5	0.0	0.0	37.5	25.0
	7	0.0	0.0	0.0	0.0	0.0	0.0	0.0	11.1	22.2	66.7
	8	0.0	0.0	0.0	0.0	0.0	11.1	0.0	0.0	27.8	61.1
	9	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	16.7	83.3

6 Conclusions and further work

This paper has empirically studied the use of ensemble techniques in combination with Learning Classifier Systems, specifically using GAssist, a Pittsburgh approach LCS. Two types of techniques are studied. The first kind of ensemble performs a Bagging-style consensus prediction, while the second one is an hierarchical ensemble intended to deal with ordinal classification domains.

These methods are not really new contributions, just variations of already existing techniques, but the experiments reported in the paper illustrate that they are very useful in combination with GAssist. The first one boots significantly the performance of GAssist on several domains, while the other one helps GAssist minimize the importance of the mis-classifications, which is an issue of concern in ordinal domains.

For future work, it would be interesting to determine how can we tweak GAssist to provide the correct models to the ensemble in order to maximize the accuracy of the consensus prediction. For the hierarchical ensemble, a possible future line is to study other policies to partition the ordinal domain into several binary sub-domains, such as the ones described in the related work section. Finally, checking the accuracy-computational cost tradeoff of these ensemble techniques would be very useful.

7 Acknowledgements

We acknowledge the support provided by the UK Engineering and Physical Sciences Research Council (EPSRC) under grants GR/T07534/01 and GR/62052/01 and the Biotechnology and Biological Sciences Research Council (BBSRC) under grant BB/C511764/1.

8 References

[1] J. Bacardit. *Pittsburgh Genetics-Based Machine Learning in the Data Mining era: Representations, generalization, and run-time*. PhD thesis, Ramon Llull University, Barcelona, Catalonia, Spain, 2004.

[2] J. Bacardit. Analysis of the initialization stage of a pittsburgh approach learning classifier system. In *GECCO 2005: Proceedings of the Genetic and Evolutionary Computation Conference*, volume 2, pages 1843–1850. ACM Press, 2005.

[3] J. Bacardit, M. Stout, N. Krasnogor, J. Hirst, and J. Blazewicz. Coordination number prediction using learning classifier systems: Performance and interpretability. In *GECCO 2006: Proceedings of the Genetic and Evolutionary Computation Conference (to appear)*, 2006.

[4] C. Blake, E. Keogh, and C. Merz. UCI repository of machine learning databases, 1998. (www.ics.uci.edu/mlearn/MLRepository.html).

[5] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[6] K. A. DeJong, W. M. Spears, and D. F. Gordon. Using genetic algorithms for concept learning. *Machine Learning*, 13(2/3):161–188, 1993.

[7] E. Frank and M. Hall. A simple approach to ordinal classification. In *Proc 12th European Conference on Machine Learning*, pages 145–156. Springer, 2001.

[8] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, pages 148–156, 1996.

[9] A. J. B. L. Bull, M. Studley and I. Whittley. On the use of rule sharing in learning classifier system ensembles. In *Proceedings of the 2005 Congress on Evolutionary Computation*, 2005.

[10] H. Liu, F. Hussain, C. L. Tam, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, 2002.

[11] X. Llorà, J. Bacardit, E. Bernadó, and I. Traus. Where to go once you have evolved a bunch of promising hypotheses? In *Advances at the frontier of Learning Classifier Systems (Volume I)*. (in press), LNAI, Springer-Verlag, 2006.

[12] C. Richardson and D. Barlow. The bottom line for prediction of residue solvent accessibility. *Protein Eng*, 12:1051–1054, 1999.

[13] various authors. Special issue on integrating multiple learned models. *Machine Learning*, 36(1-2), 1999.