Optimizing Nearest Neighbour in Random Subspaces using a Multi-Objective Genetic Algorithm

Guillaume Tremblay [1] Robert Sabourin[1] Patrick Maupin[2]
[1] École de technologie supérieure, Montréal, Canada
Guillaume.Tremblay@livia.etsmtl.ca, Robert.Sabourin@etsmtl.ca
[2] Defence Research and Development Canada, Valcartier, Canada
Patrick.Maupin@drdc-rddc.gc.ca

Abstract

In this work, the authors have evaluated almost 20 millions ensembles of classifiers generated by several methods. Trying to optimize those ensembles based on the nearest neighbours and the random subspaces paradigms, we found that the use of a diversity metric called "ambiguity" had no better positive impact than plain stochastic search.

1. Introduction

Research on ensembles of classifiers (EoC) has become an important topic in the pattern recognition community. It is now well known that, in order to achieve high recognition rate, individual classifiers belonging to EoC must yield different outputs. The key idea is that if fairly good classifiers make errors on different examples, the ensemble should be more accurate than its single best element. Diversity can take multiple forms and, in the case of voting classifiers, a measure called "ambiguity" has been proposed [4] to help the creation of high-performance EoCs. This measure represents the average rate of classifiers that disagree with the ensemble's decision.

The ambiguity a_i of the i[th] classifier on the observation k is:

$$a_i(k) = \begin{cases} 0 \text{ if } \mathrm{Class}V_i(k) = \mathrm{Class}\overline{V}_i(k) \\ 1 \text{ Otherwise} \end{cases} \quad (1)$$

where $\mathrm{Class}V_i(k)$ is the class chosen by the classifier and $\mathrm{Class}\overline{V}_i(k)$ is the output of the ensemble. The ambiguity of the EoC on a given database is:

$$\overline{A} = \frac{1}{|K| \cdot |I|} \sum_{i \in I} \sum_{k \in K} a_i(k) \quad (2)$$

where I is the set of classifiers, K is the set of examples in the database and $|I|$ and $|K|$ are the cardinality of I and K respectively.

Using a hill-climbing method, it has been shown that maximizing ambiguity in ensembles of kNN creates more accurate EoCs than if no measure of diversity is used [4]. It has also been shown that the random subspace method is efficient in creating kNN-based EoCs that have higher recognition rate than a single kNN [3]. Knowing that such ensembles have an inherent diversity, would there be any benefit in using the measure of ambiguity as an optimization criteria in order to maximize performance? Can we significantly reduce the number of kNN in the ensemble, and thus reduce the cardinality of the EoC without any loss of performance in generalization?

2. Optimization methods

In this work, we analyzed the effect of using a measure of diversity called *ambiguity* [4] in the optimization of ensembles of kNN generated by the random subspace method. The optimization is done using a multi-objective genetic algorithm and the NIST SD19 database [1]. The data consists of handwritten numerals forming a 10 classes pattern recognition problem.

Prior to the optimization, we have experimentally determined that a one-nearest neighbour classifier (k=1) in a subspace of 32 features (out of 132) was the best combination of parameters for our ensembles of kNN.

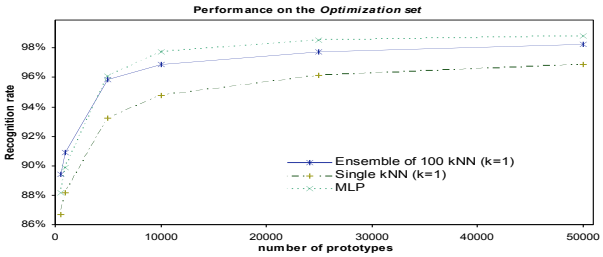

Figure 1: Accuracy evolution as the training set increases for MLP, kNN and ensemble of 100 kNN in random subspace.

We have created and trained ensembles of 100 kNN with several subsets of the NIST SD19 database (0,5, 1, 5, 10, 25, and 50 x 10^3 prototypes). For consistency, the recognition rate of these ensembles was measured on the same database containing 10,000 examples (Figure 1). The evaluation of the performance was done by taking the vote of each kNN of the ensemble. The class with the highest number of votes wins and ties are arbitrarily resolved.

For the rest of the paper, we focused on the ensemble trained with 5000 prototypes and compared a number of methods of classification based on the 100 base classifiers.

The *single kNN* (Figure 1) is used as reference and is composed of a single 1-NN classifier trained with the full feature set. A multilayer perceptron (MLP) is also used as reference. More details on that neural net can be found in [1]. The *base EoC* is made of 100 1-NN generated by the random subspace method [3], each of which having been trained with 32 randomly selected features out of the initial feature space. This base EoC provides a reference for the optimized EoC as it is composed of a non-optimized ensemble already better than the single kNN. Therefore, the search for a better ensemble consists in finding, if it exists, a subset of N classifiers out of the 100 base classifiers forming the base EoC. A better ensemble is defined as one that has a higher recognition rate. For equal recognition rates, the classifier with the lower number of classifiers is considered as the best one.

Certainly the most straightforward search method, the *ranked search* consists in sorting the base classifiers by decreasing recognition rate, then adding up these classifiers one by one to create a growing EoC. The EoC is evaluated at each size, starting with only the single best classifier, and then adding the second best to the ensemble and so on. This method creates a set of 100 EoCs from which we can chose the one we consider the better.

As with the ranked search, stochastic search consists in building 100 EoCs by adding to the ensemble individual classifiers one by one. In this case, the order in which the kNN are put in the EoC is randomly selected. Every time a kNN is added, the recognition rate of the new EoC is evaluated. The procedure can be repeated for a number of times and the final set of solutions contains the single best EoC of each size (from 1 to 100). Stochastic search allows thus a quite exhaustive sampling of the search space. It is a good way to observe the "natural" behaviour of the different measures taken here and the relation between them (recognition rate, number of classifiers and ambiguity).

All four remaining search methods use genetic algorithms (GA) and are hence guided search. The first obvious GA used is a simple GA with one objective: to maximize the recognition rate of the EoC. Each kNN is represented by a bit in a 100 bits chromosome. Turning on/off a bit selects/unselects the corresponding classifier in the EoC.

The non-dominated sorting GA (NSGA) [1] approach used here tends to maximize or minimize two or more objectives at the same time in multi-dimensional fitness spaces, avoiding thus the aggregation of metrics capturing multiple objectives into a single metric. This is achieved by sorting the evaluated solutions by Pareto fronts. This kind of algorithms produces a set of non-dominated solutions (the Pareto front, Figure 2) from which we can choose the one we consider the better according to a new criterion.

Three (3) different couples of objective functions have been used for the search with NSGA, each of these for a particular purpose. The first one is to maximize conjointly the recognition rate (F1) and the ambiguity ($F2_a$), which is precisely what this work aims to study.

A second couple of objective functions has been chosen: to minimize the error rate (F1') and to minimize the ambiguity ($F2_b$), in order to get a better understanding of the behaviour of the ambiguity as and optimization criteria. But for the formalism and some graphical representation, F1' and F1 are the same logical objective. Joint minimization of F1' and $F2_b$ may result in high margin EoCs, e.g. their average degree of agreement has good chances of being located far from the majority threshold of 50%, an ideal situation according to [2].

F1' along with minimization of the cardinality of the EoCs ($F2_c$) constitutes the last couple of objective functions. Minimizing the cardinality of the EoC is a natural objective and the most obvious way to reach it is to explicitly put pressure on it.

3. Experimental Protocol

The optimization of EoCs requires four (4) distinct databases, that is one more than when performing the optimization of a single classifier. The first data set is used to train the base classifiers, here corresponding to the kNN's memory. It is called the "*training set*", the "*training database*" or simply the "*prototypes*". A set called the "*optimization set*" is used to measure the performance of the generated EoCs. All search methods in this paper aim at minimizing recognition error on that database. Since these search methods offer the choice of several EoCs, a third database called "*validation set*" is needed to choose the final EoC in order to avoid overfitting. Finally, a *test database* is used to measure performances in generalization.

All data sets come from NIST SD19. The test database has 60089 examples (hsf_7). The *optimization set* and the *validation set* each contain 10,000 examples. The size of the training set is a parameter that could eventually vary but that is fixed at 5000 examples for the current study. All samples related to the optimization, validation and training datasets were extracted from hsf_0-3. In this case, each class is composed of the same number of samples.

Experiments are divided into two categories. The first category regroups the reference techniques that produce a single solution (classifier or EoC), e.g.: the simple kNN, the MLP, the base EoC and the ranked search. The second is a set of techniques rather producing a set of solutions, e.g.: the stochastic search, the NSGA-based algorithm and the simple GA.

Several MLPs have been trained with the *training set* while the *optimization set* was used to avoid overfitting. Each of these neural nets had a different number of units in the hidden layer according to the methodology

described in [1]. The *validation set* was used to choose the best configuration.

The GA-based experiments all share the same setup: 128 individuals in the population and 1000 generations, which means that 128,000 EoCs are evaluated in each of these experiments. The stochastic search produces 100 EoCs at each run. To compare with the GA-based methods, 1280 runs have been conducted for the same total of 128,000 evaluated EoCs.

All kNN are trained with the training set. The ranking for the ranked search and the recognition rate for the other methods are calculated on the optimization set. When the algorithm stops, the ensemble of solutions proposed by the algorithm is evaluated using the validation database. The EoC with the highest performance on that set is chosen as the best one found with a particular method. This EoC is then evaluated on the test set, giving a final score that can be compared with the score of other EoC obtained by different search methods. At every stage performance, ambiguity and cardinality of each EoC are recorded to allow further analysis of the results.

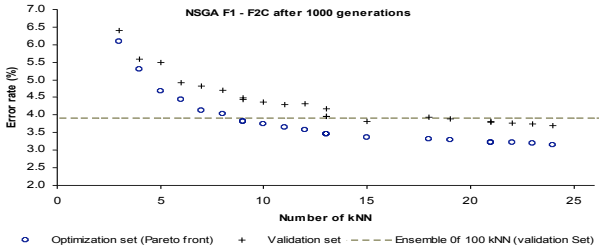

Figure 2: Example of a Pareto front after 1000 generations (NSGA F1-F2$_c$).

The set of solutions for the stochastic search is made of the best EoC evaluated for all cardinalities from 1 to 100. NSGA produces a set of solutions directly from the Pareto front while the whole population of the last generation is considered as the solution set for the single GA approach.

Each experiment was replicated 30 times, giving thus 30 optimized EoCs according to each of the 5 search methods. These replications allow further statistical analysis of the behaviour and expected performance of each search method. In the end, 19.2 x10^6 EoCs where evaluated (128 x 10^3 x 5 x 30).

4. Results

As it has already been demonstrated [3, 4], kNN-based EoCs have higher recognition rates than simple kNN (Figure 3 and Table 1). The single kNN recognition rate - slightly higher than 93% on all data set - is inferior to the recognition rate of other searches. The highest improvement in the recognition rate occurs between the single kNN and the ensemble of 100 kNN. In other words, it is the transition from a one-classifier technique to an EoC technique that gives the most important gain.

Table 1: Recognition rates of reference methods

	Optimization	Validation	Test
Single kNN (132 feat.)	93,23	93,33	93,34
MLP (132 feat.)	96,11	95,91	95,27
Ensemble of 100kNN	95,79	96,09	96,28
Ranking (76 kNN)	95,93	96,15	96,26

The biggest impact of optimizing the base EoC is in the reduction of its cardinality (Figure 4) though it is possible to slightly improve the recognition rate in generalization (test set). GA-based methods have the highest recognition rates on the *optimization set* (in memorization) but that performance drops off on the other sets (in generalization).

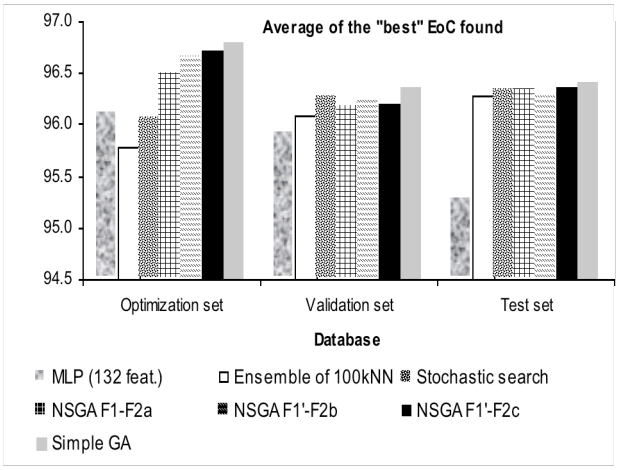

Figure 3: Accuracy of the best ensemble found for all studied search methods on the 3 data sets (average).

A Student's two-tailed t-test shows that the difference between the average results on the simple GA and results of other methods is significant at a confidence level of 95%. Consequently, it is safe to say that the simple GA gives the most accurate EoCs on all data sets; the EoCs found with that method has an average recognition rate of 96.41% on test set. The use of the genetic operators is therefore more efficient at finding good solutions than the exhaustive stochastic search or the use of a heuristic like the ranking. The impact on the cardinality is even more important, with an average of 34.66 classifiers against 62.50 for the stochastic search and 76 for the ranked search.

Further Student's t-tests show that the stochastic search, NSGA F1-F2$_a$ and NSGA F1'-F2$_c$ produce EoCs that are not significantly different as far as the recognition rate is concerned. While their average recognition rate is slightly under the one obtained with the simple GA, EoCs generated by NSGA F1'-F2$_c$ have fewer classifiers than others EoCs (average of 18.87).

The loss in performance (96.35%) on the test set might be considered as very small in comparison of the

reduction of cardinality. Student's t-tests showed no difference between the average cardinality of EoCs generated by NSGA F1'-$F2_b$ and NSGA F1'-$F2_c$.

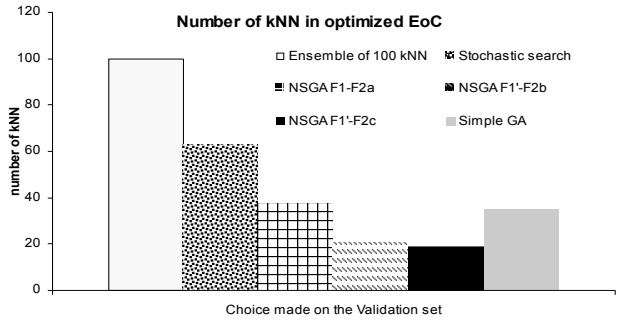

Figure 4: Complexity of the best ensemble found for the studied search methods (average).

The use of a measure of ambiguity did not help the search of better EoCs. The intrinsic diversity of the random subspaces-based kNNs yields ambiguity measures that are very difficult to either maximize or minimize (Figure 5). Moreover the EoCs obtained by the 6 search methods all have more or less the same recognition rates.

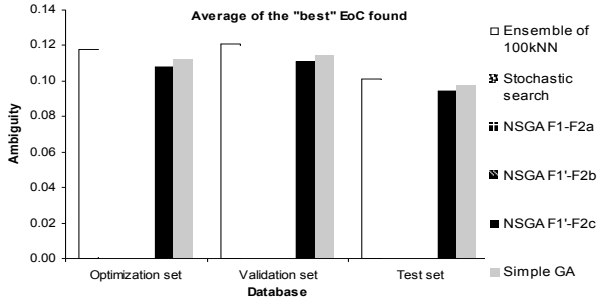

Figure 5: Ambiguity of the best ensemble found for all studied search methods on the 3 data sets (average).

It is interesting to note that while the ranked search did not give the best results either in term of complexity nor performance, the method is so simple to apply that it might well be of some use in practice. As an example, Table 2 shows recognition rates on the test database of the ranked search for the 15, 20, and 25 best classifiers.

Table 2: Recognition rates on test set for the 15, 20 and 25 best classifiers using the ranked search method

Number of kNN	15	20	25
Recognition rate	96.00	96.12	96.16

At the cost of a fraction of a percent in the recognition rate, one can chose arbitrary cardinality using the easiest method to deploy.

5. Conclusion

The simple GA is the best search method we found to maximize the performance of EoCs (Figure 6). The median recognition rate of its classifiers is higher or equal to the ones in the first three quartiles of the other search techniques and more than 75% of its solutions are better than the median solution of other searches. Its worst case is also better than all other worst cases. $F2_b$ gives the worst average results and has the worst case with the smallest recognition rate. Other techniques have equivalent average recognition rates though the variations around it differ.

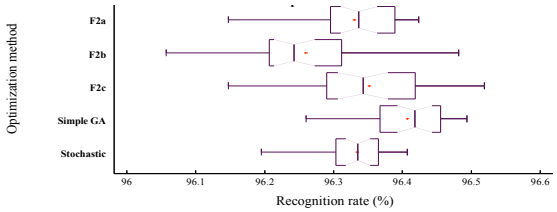

Figure 6: Results of 30 replications of optimized EoC on the test database. Left and right whiskers represent 1st and 4th quartiles respectively. The vertical line is the median and separates 2nd and 3rd quartiles (the box encloses 50% of the EoCs). Plus sign is the mean.

The use of ambiguity did not help much the creation of EoCs and might not be appropriated for ensemble of kNN generated by the random subspaces method. The simplest way to optimize such ensemble might be the use of a ranking heuristic at the cost of a small loss in performance. Future work will include other diversity measures [5] and other sizes of training database.

6. Acknowledgements

This research is supported by DRDC-Valcartier under the contract W7701-024425/001/QCA.

7. References

[1] L. S. Oliveira, R. Sabourin, F. Bortolozzi, C. Y. Suen, A Methodology for Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit String Recognition, *IJPRAI,* vol. 17:6, 2003, pp. 903-929

[2] D. Ruta, B. Gabrys, A Theorical Analysis of the Limits of Majority Voting Errors for Multiple Classifier Systems, *Pattern Analysis & Applications,* vol. 5, 2002, pp. 333-350

[3] T. K. Ho, Nearest neighbors in random subspaces, *Proceedings of the 2nd Int'l Workshop on Statistical Techniques in Pattern Recognition*, Sydney, Australia, August 1998, pp. 640-648

[4] G. Zenobi, P. Cunningham, Using Diversity in Preparing Ensembles of Classifiers Based on Different Feature Subsets to Minimize Generalization Error, *ECML*, Springer Verlag, 2001, pp. 576-587

[5] L. I. Kuncheva, C. J. Whitaker, Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy, *Machine Learning,* vol. 51, 2003, pp. 181-207