Open access • Proceedings Article • DOI:10.1109/ICPR.2008.4761264

Feature selection combining genetic algorithm and Adaboost classifiers
— **Source link** ↗

Hassan Chouaib, Oriol Ramos Terrades, Salvatore Tabbone, Florence Cloppet ...+1 more authors

Institutions: Paris Descartes University

Related papers:

- Genetic Algorithm Based Optimization for AdaBoost

- A Hybrid Ensemble Algorithm Combining AdaBoost and Genetic Algorithm for Cancer Classification with Gene Expression Data

- An Improved Adaboost Algorithm Based on Uncertain Functions

- Joint feature selection and hierarchical classifier design

- Toward integrating feature selection algorithms for classification and clustering

Share this paper: 🅕 🐦 in ✉

Feature selection combining genetic algorithm and Adaboost classifiers

Hassan Chouaib, Oriol Ramos Terrades, Salvatore Tabbone, Florence Cloppet, Nicole Vincent

Feature selection combining genetic algorithm and Adaboost classifiers

H. Chouaib[1], O. Ramos Terrades[2], S. Tabbone[2], F. Cloppet[1], N. Vincent[1]

[1]Laboratoire CRIP5(EA 2517), Université Paris Descartes, France
[2]LORIA-Université Nancy 2, Campus scientique BP 239 Nancy,France

E-mail: {chouaib,cloppet,vincent}@math-info.univ-paris5.fr,{ramos,tabbone}@loria.fr

Abstract

This paper presents a fast method using simple genetic algorithms (GAs) for features selection. Unlike traditional approaches using GAs, we have used the combination of Adaboost classifiers to evaluate an individual of the population. So, the fitness function we have used is defined by the error rate of this combination. This approach has been implemented and tested on the MNIST database and the results confirm the effectiveness and the robustness of the proposed approach.

1 Introduction

With the rapid advancement of computer and database technologies, datasets with hundreds and thousands of variables, or features, are now ubiquitous in pattern recognition, data mining and machine learning applications. Thus, feature selection has become the focus of many research areas in recent years and it consists in selecting a subset of few features being the most representative for a particular application.

Genetic Algorithms (GAs), used for feature selection, and based on the wrapper method [4], need a classifier (SVM, Neural network, Near-Neighbour..) to evaluate each individual of the population [1, 6, 11]. But training classifiers at each iteration of GA is too much time consuming.

In this paper, we propose a new feature selection method, based on GA, which avoid classifier training at each iteration of the GA. Thus, a classifier is trained before running the GA but the evaluation of the individuals is done at each iteration using always the same classifier. The fitness function is based on Adaboost classifiers associated with the features. More precisely, an Adaboost classifier is trained for each feature before launching the GA for feature selection. Then, we combine (a mean operator) the Adaboost classifiers selected

at each GA iteration similarly to the method proposed in [12].

The remainder of the paper is structured as follows: in Section 2, we briefly motivate our method of feature selection. GAs and Adaboost classifiers are reviewed in Section 3 and in Section 4, respectively. Then, we introduce our feature selection method based on GA, in Section 5. Section 6 is devoted to experimental results and we draw the conclusion and the perspectives of this work in Section 7.

2 Feature selection

The curse of dimensionality is a well-known problem in pattern recognition application and several research efforts have been done in reducing the dimensionality of feature vectors. Irrelevant and redundant features may negatively affect in classifiers accuracy. If we reduce the number of features, then we make the classification models simpler and easier to understand; we decrease the cost of stocking data; and we also increase the performance of indexing methods when they are applied to feature vectors. Indeed, three goals are stated in [5] to perform a feature selection:

- Reduce the cost of feature extraction.

- Improve precision during classification.

- Improve the confidence of classifier performance.

Feature selection algorithms fall into three categories: Embedded, Filters and Wrappers methods. Embedded methods perform feature selection in the process of training and are usually specific to given learning machines. Filters select subsets of features as a preprocessing step, independently of the chosen predictor. Wrappers use learning machines of interest as a black box to score subsets of features according to their predictive power.

3 Genetic Algorithms (GAs)

GAs belong to a group of methods, called evolutionary algorithms, that have been applied to feature selection with different degrees of success [9]. Besides, GAs have been studied and proven effective in conjunction with various classifiers, including nearest neighbours and neural networks [1].

GAs are optimization procedures inspired by the mechanisms of natural selection. In general, GAs start with an initial set of random solutions called *population* [3].

A GA generally has four components. A *population* of individuals where each individual in the population represents a possible solution; a *fitness function* which is an evaluation function by which we can tell if an individual is a good solution or not; a *selection function* which decides how to pick good individuals from the current population for creating the next generation; and *genetic operators* such as crossover and mutation which explore new regions of search space while keeping some of the current information at the same time.

Each individual in the population, representing a solution to the problem, is called a *chromosome*. Chromosomes represent candidate solutions to the optimization problem being solved. In GAs, chromosomes are typically represented by bit binary vectors and the resulting search space corresponds to a high dimensional boolean space. It is assumed that the quality of each candidate solution can be evaluated using the fitness function.

4 Adaboost

Boosting algorithms increase the performance of *weak* binary classifiers by reinforcing training on misclassified samples. In particular, Adaboost (Adaptative boosting) is a widely used boosting algorithm that weights a set of weak classifiers according to a function of the classification error [2]. Thereby, the final classifier is given by:

$$h(x) = \begin{cases} 1 & \text{if } \sum_{i=1}^{t} \alpha_t h_t \geq \text{threshold} \\ 0 & \text{otherwise} \end{cases}$$

Where 1 means that the sample has been classified as belonging to the class to be identified.

5 Proposed method

In this section, we introduce the proposed method. A large set of features is assumed to be available in order to characterize a given class. This method begins by training an Adaboost classifier for each feature to be used in the fitness function. This part is independent of the GA and then is performed only once. Then, we apply GAs several times to find an optimal subset of features. Thus, our feature selection algorithm is decomposed in two steps:

Step 1 : Train Adaboost classifier for each feature.

Step 2 : Use the GA to select the best classifier or feature subset.

In what follows, we explain how to construct the fitness function (step 1) and how to train classifiers and select the optimal features (step 2).

5.1 Step 1: Fitness function

Fitness function plays the most important role in genetic search. This function has to evaluate the goodness of each chromosome in a population. Thus, the input of the fitness function is a chromosome and it returns a numerical evaluation representing the goodness of the feature subset.

In this context, a chromosome is a n dimensional binary vector, where n is the total number of features. If the i-th bit of the vector is 1, then the i-th feature is included in the subset. On the contrary, if the i-th is 0, the feature is not included. The fitness function is determined for each chromosome in the population. More specifically, the fitness function is the error rate of the mean of Adaboost classifiers selected by each chromosome. Let note H be: $H = \frac{1}{|I|} \sum_{i \in I} h_i$

where I is the set of selected features and $|I|$ the size of I. Then, the Fitness function is defined: $Fitness = Error(H)$.

For example, let $X = 1101001$, be a chromosome. Then $I = \{1, 2, 4, 7\}$, $|I| = 4$ and the mean is computed on classifiers: h_1, h_2, h_4 and h_7. Therefore, the goal of our method is to select the optimal chromosome minimizing this *Fitness* function, i.e. minimizing the classifier error.

5.2 Step 2: Feature Selection

The initial population is randomly generated. However, we add a singular chromosome composed of all features in order to ensure that the selected features perform better than the whole features.

For each generation (iteration) of the GA, each chromosome is evaluated using the $Fitness$ function. Population evaluation, by the means of the Fitness function, is a critical step of the selection process since offspring in the GA for the next generation are determined by

the fitness values of the current population. The generational process ends when the termination criterion is satisfied –in our case, the number of generations–. The selected features correspond to the best individual in the last generation.

6 Experiments and discussion

The reported results are obtained applying our method to the MNIST dataset and to three well-known descriptors. Besides, we have compared our method to related approaches: the Maximun relevant Minimum redundancy method [7] and a GA+wrapper configuration [1, 6]. Our aim is not to reach the best recognition system but to show the efficiency of the feature selection process.

To run our method three parameters have to be set experimentally. The initial GA population is composed of 200 chromosomes, the maximal number of generations is set to 50 and the maximal number of iterations for the Adaboost classifier to 50.

We have used the MNIST training dataset to evaluate our method. This dataset contains about $60,000$ handwritten digit images of 28×28 distributed in ten classes –corresponding to digits: $0,\dots,9$–. For each class, we have created three sets, namely A, B and C, composed of $1,000$ samples, randomly chosen. Set A is used for training Adaboost (step 1). Sets B and C are used, respectively, for training and testing. Furthermore, we have permuted the role played by each set in order to apply a cross-validation scheme.

All the selection results in the sections to follow have been obtained by applying the GA serval times .

6.1 Descriptors and classifiers

We have computed on these three sets three different descriptors: Zernike, $\mathcal{R}$-signature and pixels. Before computing each descriptor, we have extracted the bounding box of each digit and we have resized the image to a 32×32 image. 47 Zernike descriptors (ZER) are composed of the first twelve Zernike moments [8]. The $\mathcal{R}$-signature (RS) is a descriptor based on the Radon transform proposed in [10]. Finally, the pixels descriptor (pixels) simply consists to take each pixel as a feature in the MNIST images.

Three different one versus all classifiers: SVM, Adaboost and K-NN are used in order to make the analysis of results more independent of the classifier used. $1,000$ negatives samples from training and testing set have been randomly chosen among sets B and C, respectively.

Table 1. Number of features selected

Descriptor	Initial features	Selected features
RS	180	110
ZER	47	36
Pixels	1024	765
RS + ZER	$180 + 47 = 227$	$108 + 32 = 140$
RS + Pixels	$180 + 1024 = 1204$	$105 + 688 = 793$
ZER + Pixels	$47 + 1024 = 1071$	$36 + 692 = 728$

6.2 Results

Results from two sets of experiments are reported. The first set of experiments show the stability of our method with respect to different configurations. The second set of experiments has been addressed to state the performance of our method compared to related feature selection algorithms.

For the first set of experiments, several descriptors, combination of descriptors (RS+ZER, RS+Pixels and ZER + Pixels) and classifiers have been used. It can be noticed that we consider a set mixing two families of features. The selection contains features from both families and respects almost the same ratio as in the initial set. Table 1 shows the number of selected features for each descriptor. For the majority of descriptors we can notice that the number of features is decreased more than 35%. Indeed, the performance of the three classifiers has essentially been unchanged after applying our method –see Table 2, confirming the robustness of the approach.

The second set of experiments is devoted to compare our method to other related feature selection methods. The maximum relevant minimum redundancy method (MRMR) permit to fix the number of features to be selected [7]. Thus, we have chosen the mean number of features selected, for each descriptor, by our method in the first experiment. We have also compared to a GA+Wrapper configuration for feature selection. As it has been explained in the Introduction section, when GAs are used for feature selection the Wrapper strategy is the most commonly used. Thus, we have used the Adaboost as wrapper in order to make comparison to our method easier. Results can be observed in Table 3. According to these results, on the one hand, an improvement is done by our method comparing to the MrMr approach. On the other hand, the performance of our method is quite similar to GA+Adaboost. However, our method is faster (in the worst case 20 times faster and in the best case 50 times, see Table 4).

Table 2. Recognition rates for three different classifiers: SVM, Adaboost and K-NN (K=5), trainned with all features and using only the features selected by our selected method.

Descriptor	SVM		AdaBoost		5-NN	
	ALL	Selected	ALL	Selected	ALL	Selected
RS	75.06	75.10	81.62	81.54	84.55	84.52
ZER	84.00	82.28	81.11	80.78	88.28	87.58
Pixels	97.72	97.70	94.35	94.43	97.26	97.28
RS+ZER	86.32	86.55	81.02	82.55	88.28	87.65
RS+Pixels	97.68	97.66	94.59	94.70	97.26	97.36
ZER+Pixels	97.77	97.77	94.32	94.55	88.36	87.90

Table 3. Comparison with GA+Adaboost (wrapper) and MrMr

Descriptor	GA+Adaboost	MrMr	Our method
RS	75.10	71.78	75.16
ZER	83.04	81.00	82.24
Pixels	97.95	96.95	97.712

Table 4. Relative time execution between GA+Adaboost and our approach.

Descriptor	GA+Adaboost	Our method
RS	78.125	2.656
ZER	51.875	1
Pixels	260	11.75

7 Conclusion and perspectives

A new approach for feature selection in classification problems using GAs is introduced in this paper. The fitness function used is the combination of several Adaboost classifiers.

Unlike the traditional selections methods using GAs, and based on the wrapper method we have proposed a new selection approach using a simple GA and a *priori* classifiers. Besides, the evaluation of individuals is done by combination of simple classifiers trained by Adaboost for each feature. In this perspective, the complexity of the approach is substantially reduced.

Our cross validation scheme performed on a MNIST dataset shows that similar results can be obtained using about 35% less features in multi-class context and 75% less when a two class problem is considered. Therefore, the robustness of the proposed approach is also confirmed.

Future works will be devoted to improve the combination output of Adaboost classifiers selected at each GA iteration. Thus, a weighted sum will be considered, where each weight will depend on the fitness error.

References

[1] A. A. Altun and N. Allahverdi. Neural network based recognition by using genetic algorithm for feature selection of enhanced fingerprints. In *ICANNGA*, 2007.

[2] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[3] J. H. Holland. Adaptation in natural and artificial systems. *Ann Arbor, MI, Univ of Michigan Press*, 1975.

[4] G. H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, 1994.

[5] M. Kudo, P. Somol, P. Pudil, S. M., and S. J. Comparison of classifier-specific feature selection algorithms. In *SSPR*, pages 677–686, 2000.

[6] L.Oliveira, R.Sabourin, F.Bortolozzi, and C. Suen. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *International Journal of Pattern Recognition*, 03.

[7] F. Long and C. Ding. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(8):1226–1238, 2005.

[8] R. Prokop and A.P. Reeves a survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP*, 54(5), 1992.

[9] W. Siedlecki and Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, pages 335–347, 1989.

[10] S. Tabbone and L. Wendling. Recognition of symbols in grey level line drawings from an adaptation of the radon transform. In *In Proceedings of 17th ICPR, Cambridge (UK)*, pages 570–573, 2004.

[11] F. Tan, X. Fu, Y. Zhang, and A. G. Bourgeois. A genetic algorithm-based method for feature subset selection. *Soft Computing*, 12(2):111–120, 2007.

[12] X.-C. Yin, C.-P. Liu, and Z. Han. Feature combination using boosting. *PRL*, 26(4):2195–2205, 2005.