Generic Feature Selection and Document Processing

H.Chouaib[1], N.Vincent[1], F.Cloppet[1], S.Tabbone[2]
[1]Laboratoire CRIP5(EA 2517),Université Paris Descartes, France
[2]LORIA-Université Nancy 2, Campus scientique BP 239 Nancy,France
{hassan.chouaib,florence.cloppet,nicole.vincent}@mi.parisdescartes.fr
tabbone @loria.fr

Abstract

This paper presents a generic features selection method and its applications on some document analysis problems. The method is based on a genetic algorithm (GA), whose fitness function is defined by combining Adaboot classifiers associated with each feature.

Our method is not linked to a classifier achieving the final recognition task; we have used a combination of weak classifiers to evaluate a subset of features. So we select features that can further be used in the most appropriate classifiers.

This method has been tested on three applications: Drop caps classification, handwritten digits recognition and text detection. The results show the efficiency and robustness of the proposed approach.

1 Introduction

When dealing with data, for example in a pattern recognition context, it is essential to analyze the properties of the data. In fact, the efficiency of a method often relies more on the quality of features chosen by the expert than on the classifier type used during the recognition process. In order to choose the best features, a complete analysis should be done and achieved on a very large amount of training data. This is seldom done in real applications. Features are often tried and empirically chosen. A very common idea is that you get more information when using a very large number of features. In fact some non discriminative features or features too sensitive to noise can decrease the recognition rate. Besides, a very particular feature can be efficient to discriminate some shapes occurring not too often. Rather than processing with a large number of trials, we think a selection step would be more efficient. It would enable not to limit the number of first considered features, while features in a reasonable number would be computed in the final system. This can explain feature selection has been a hot topic research in recent years.

Among all these studies, one of the most common is based on Genetic Algorithms (GAs). They have proven to be efficient for feature selection. Most often, the approach is based on the wrapper method [4]. It needs a classifier (SVM, Neural network, Near-Neighbour..) to evaluate each individual of the population [1, 5, 10] and a training phase at each iteration of GA which is very time consuming.

We propose a new feature selection method, based on GA, which avoids these training steps. Thus, classifiers are 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, the Adaboost classifiers selected at each GA iteration are combined similarly to the method proposed in [11].

In this paper, our generic feature selection method is experimented on different document analysis problems relying on a feature vector. In section 2, GA and Adaboost are reviewed. We then recall the proposed selection method and stress on the interesting points, particularly the speed of the process compared with more classical approaches. Section 3 is devoted to some applications dealing at different levels of document observation. The first is at character level , the second one deals with images and the third one works at the page level in order to extract text parts.

2 Feature selection

The main aim of feature selection is to determine a minimal feature subset while keeping a suitably high accuracy in original features representation. In many real contexts, feature selection is needed due to the large amount of noise and the irrelevant or misleading features. Before describing our method involving GAs and Adaboost classifiers, the key ideas they rely on, are summarized.

978-0-7695-3725-2/09 $25.00 © 2009 IEEE
DOI 10.1109/ICDAR.2009.200

2.1 Genetic algorithm and selection

GAs belong to a group of methods, called evolutionary algorithms, that have been applied to feature selection [8]. 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 natural selection mechanisms. In general, GAs start with an initial set of random potential solutions called *population* [3].

A GA generally has four components. A *population* of individuals where each individual in the population represents a potential solution; a *fitness function* which is an evaluation function. It enables to decide whether an individual is a good solution or not. A *selection function* indicates how to pick good individuals from the current population for creating the next generation; and *genetic operators* such as crossover and mutation operators which explore new regions of search space.

Each individual in the population, representing a solution to the problem, is called a *chromosome*. Chromosomes represent candidate solutions to the optimization problem to be 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.

2.2 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 (α_t) a set of weak classifiers (h_t) according to the classification error. 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.

2.3 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, a GA is applied several times to find an optimal subset of features.

2.3.1 selection process

The process of our feature selection method [2] is shown in Figure 1. It is composed of two steps: Train Adaboost classifier for each feature and use a GA to select the best chromosome (or feature subset).

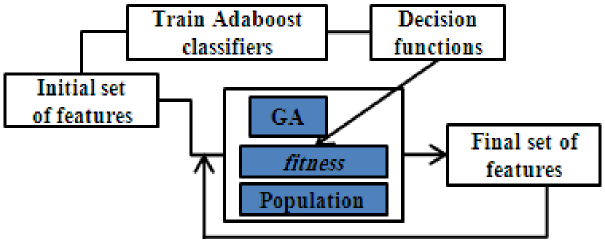

Figure 1. Selection process

Instead of selecting features that can be computed for any shape we are first adapting the features to the training set in order to increase the discriminating power of each feature in the context. This is done by building classifiers based on each feature and on an Adaboost process.

At the end of the first step, each Adaboost classifier associated with a feature, relies on the sign of a decision function. In Adaboost, the decision function defined is a linear combination of weak classifiers. This set of decision functions enables to define the fitness function of the genetic algorithm. Then, the GA is applied iteratively, begining with the initial set of features, and then on the new population built from the previously selected subset of features. The best features subset is selected at the last iteration process. More precisely, the selection process stops when the recognition rate on a validation database begins to decrease.

2.3.2 Fitness function

Fitness function is one of the most important part in genetic algorithm. This function evaluates the quality of each individual in a population, that is a feature subset.

In this context, a chromosome (individual) 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. It is calculated as the error rate of a classifier, the decision function of which is the mean value of Adaboost classifiers decision functions, present in the chromosome (computed in the first step). CL is the classifier of a chromosom:

$$CL = \frac{1}{|I|} \sum_{i \in I} H_i = \frac{1}{|I|} \sum_{i \in I} \sum_k \alpha_{ki} h_{ki}$$

where H_i is an Adaboost classifier trained for the the i-th feature, I is the set of selected features and $|I|$ the size of I. Then, the Fitness function is defined:

$$Fitness = Error(CL).$$

For example, let $X = 10010011$, be a chromosome. Then $I = \{1, 4, 7, 8\}$, $|I| = 4$ and the mean is computed on classifiers: H_1, H_4, H_7 and H_8.

2.3.3 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.

The iterations in the feature selection process end when the termination criterion is satisfied. In our case, the criterion is the recognition rate associated with the best individual in the current GA last generation. The iterative process ends up when this rate decreases (see Figure 2) and the selected features correspond to the best individual in the penultimate iteration.

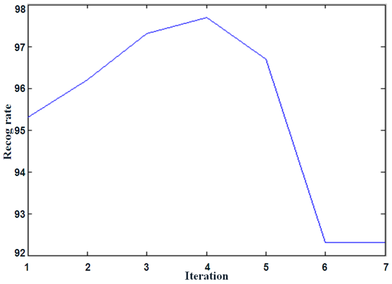

Figure 2. Recognition rate on validation database

The feature selection method we have proposed is dealing with binary classification problem. To extend it to multi classes problems, N one vs all training databases are built where N is the number of classes. Our method is applied on each database and N subsets of features are extracted. The final subset to deal with the multiclasses problem is the union of all selected subsets.

3 Application

The three applications presented in this section use the same original set of features. They are based on the possible patterns included in a 2x3 mask of pixels. With 256 grey levels the number of possible patterns would be 256^6, that is a too important number. To reduce the number of patterns involved in the application, the grey levels scale was divided into a relatively small number of classes. Thus, a quantization was applied on the original image to obtain a new 3 grey levels image. In this new image, there are 729 potential patterns ($3^6 = 729$) and the rank of each pattern is computed. This set of features makes the first descriptor. We call it the pattern descriptor.

To show the efficiency of our selection method, it has been tested on the three following applications: handwritten digits recognition, dropcaps classification and text detection in ancient documents.

3.1 Handwritten digits recognition

Various methods were proposed to solve the handwritten digits regonition problem. In our case the goal is not to find the most powerful system of recognition but to show the power of our feature selection method and how it can be applied in this kind of problems.

In this application, the MNIST database was used. iT contains about 70 000 handwritten digit images of 28 × 28 distributed in ten classes, 60 000 handwritten digits in the training set and 10,000 handwritten digits in the test set.

We have computed in this database five desciptors, discribed as folows.The first one is the pattern descriptor described previously, the second one is similar to the first but we have applied another quantization on the orignal image to obtain 2 gray levels images and 3x3 mask. The three others are classical descriptors: Zernike, $\mathcal{R}$-signature and pixels. 47 Zernike descriptors (ZER) are composed of the first twelve Zernike moments [7]. The $\mathcal{R}$-signature (RS) is a descriptor based on the Radon transform proposed in [9]. Finally, the pixels descriptor (pixels) simply consists to take each pixel as a feature in the MNIST images.

As our aim is to drastically decrease the number of features while keeping a "good" recognition rate we focus in Table 1 on the variations of these values.

Table 1. Variation of Recognition rate and feature number before and after selection

Descriptor	Recog rate %	Features %
RS	+0.04	-34
ZER	-1.7	-23.4
Pixels	-0.02	-25.3
2x3 mask + 3 means	-1.7	-27.5
3x3 mask+ 2 means	+0.04	-21.9

We can notice that many features have been kept, indeed there are 10 classes. The reduction has been done for each one vs all problem (Table 2) and in fact features are specialized for each class.

Table 2. Variation of Recognition rate and feature number for each class using 3x3 pattern

Class$_i$	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**All**
Recog rate %	-0.2	-0.5	-0.1	+0.5	+2.5	-0.68	-1.2	-2	-0.9	+2.1	+0.04
Feature Number %	-87.5	-98.24	-90.4	-63.08	-88.47	-97.46	-64.06	-88.86	-60.74	-89.64	-21.9

3.2 Dropcaps classification

The second application is about the dropcaps classification. This database is coming from the **C**entre d'**E**tude **S**upérieur de la **R**enaissance of Tours. In this database there are three styles of dropcaps that are shown in Figure 3.

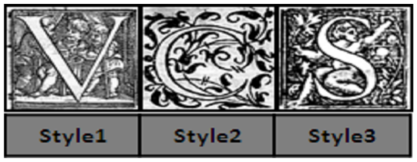

Figure 3. Styles of dropcaps

A 1-NN classifier was chosen for the classification step and the pattern descriptor previously defined was used. We have compared the recognition rates using all features (729 features) and using the features selected by our method. It selects 80 features at the second iteration and 6 features at the third. The results are shown in Table3.

Table 3. Recognition rate before and after selection

Style1	Style2	Style3	Features	GA
100	100	100	729	-
100	100	100	80	Iteration2
100	100	100	6	Iteartion3

Our results were compared to those obtained with another method proposed in [6]. This method uses seven features to classify the drop caps. These features are based on the Zipf law. They are in fact factors built from the all set of initial features. The first six features are shown in Figure 4. Features S_1, S_2 and S_3 correspond to 3 slopes. B_1, B_2 and B_3 represent the abscissa of these three points. The seventh is the slope of the inverse Zipf graph's line.

The results are shown in the Table 4. According to these results, an improvement is done by using the six features selected by our method compared to the other approach.

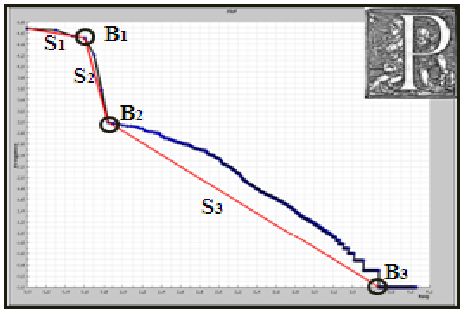

Figure 4. Features extracted using Zipf grapf

Table 4. Comparison between our method and Zipf method

	Style1	Style2	Style3	Features
Our method	100	100	100	6
Zipf method	100	95	100	7

The use of the rank of pattern as a feature helps to keep information about the used patterns. Thus, to get an interpretation of the selection process, we have analysed the six selected features (shown in Figure 5) by studying the mean of rank for each selected pattern for each style.

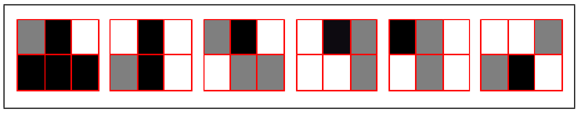

Figure 5. Selected pattern

These results show that most of the selected patterns have high rank. To localize the seven selected features on the Zipf graph we have computed the Log of rank and we have found that the selected features are in the colored region as shown on Figure 6, by studying the mean of rank for each selected pattern for each style.The high value of rank indicates that the most relevant patterns are relatively rare.

3.3 Text extraction

The pattern descriptor previously defined is also used

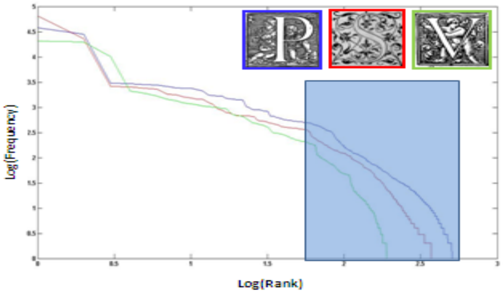

Figure 6. Feature Localisation on Zipf graph

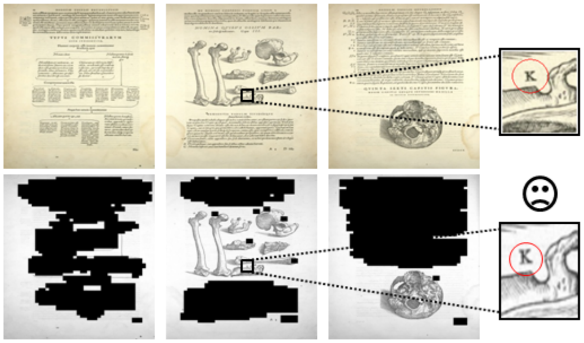

Figure 7. Detection results with 729 features

for text detection in ancient documents.The image is divided into small imagettes of size 32x32 and each of them is classified as text or non-text. This simple idea was tested using 50 pages extracted from an ancient book of Andreas Vesalius (1514-1564). These pages were scanned by the "**C**entre d'**E**tude **S**upérieur de la **R**enaissance" of Tours.The Figure 7 shows an example of detection for some pages.

Our feature selection method was applied and at the second iteration, 89 features were selected from 729 and the Figure 8 shows the results of detection after selection.

The results show that the visual quality of detection has essentially been unchanged after applying our feature selection method.

4 Conclusion

To solve the problem of feature selection, we have introduced an approach that takes into account not only the feature as input of the GA but also some knowledge about the data. This knowledge is embedded in the Adaboost classifiers that in fact replace the features in the selection process.

The flexibility of the method enables to apply it to problems with various specificities. In the three presented applications we have reduced the number of features and the recognition rates are improved.

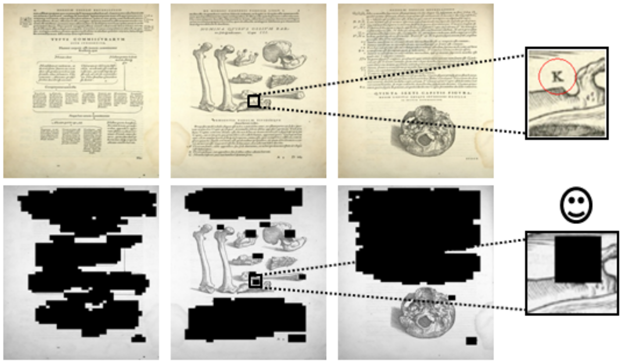

Figure 8. After selection using 89 features

We intend to peruse our work in order to improve the segmentation of document images by increasing the number of classes (text, graphic, image . . .). Thus an attributed graph could be associated with the page layout and be included in a page browsing system.

References

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

[2] H.Chouaib, S.Tabbone, O.Ramos, F. Cloppet, and N.Vincent. Feature selection combining genetic algorithm and adaboost classifiers. In *ICPR*, 2008.

[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 *ICML*, 1994.

[5] L.Oliveira, R.Sabourin, F.Bortolozzi, and C. Suen. A methodology for feature selection using multi-objective genetic algorithms for handwritten digit string recognition. *IJPR*, 03.

[6] R. Pareti and N.Vincent. Ancient letters indexing. In *ICPR*, 2006.

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

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

[9] S. Tabbone and L. Wendling. Recognition of symbols in grey level line drawings from an adaptation of the radon transform. In *Proceedings of 17th ICPR*, 2004.

[10] 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.

[11] Y. X-C., L. C-P., and H. Z. Feature combination using boosting. *PRL*, 26(4):2195–2205, 2005.