A Genetic Algorithm-based Approach for Selecting Input Variables and Setting Relevant Network Parameters of a SOM-based Classifier

SILVIA CATENI et al: A GENETIC ALGORITHM-BASED APPROACH FOR SELECTING INPUT . . . . .

Abstract — Variable selection is an important task in machine learning and data mining applications. In many real world problems a huge volume of data is often available which corresponds to a large number of variables. When developing a model for classification, clustering or other applications, the search of an optimal subset of relevant input variables is crucial. In this paper an automatic variable selection method, which combines Genetic Algorithms and Self Organizing Maps, is proposed for classification purposes. The Genetic Algorithm is used to select the most relevant input variables and to set some relevant parameters of a classifier implemented through a Self Organizing Map. This method has been tested with several datasets belonging to the UCI repository. The results of the tests are presented and discussed in this paper. The proposed approach provides a good classification accuracy and contributes to the comprehension of the phenomenon under consideration.

Keywords - variable selection, feature selection, genetic algorithms, Self Organizing Maps, classification

I. INTRODUCTION

Input variables selection is an important task within the whole process of developing a multivariate model through the exploration of a database that gathers data describing a particular phenomenon or process.

Variable selection is an important data pre-processing step in several fields, such as machine learning [1] [2], data mining [3], medical data [4] and pattern recognition [5, 6].

The main aim of this work is to select a subset of variables among all the available ones to be fed as input to a SOM-based classifier. Such variables should hold the highest informative content with respect to the input-output mapping which is to be developed. An effective selection process results in the creation of a well performing classifier [7]. On the contrary, an incorrect selection of the model input variables negatively affects the performance of the system. Variable selection should be able to point out all the variables affecting the phenomenon under consideration. Non relevant variables should not be selected. On one hand, the missed selection of one or more relevant variables prevents the model to represent some salient input-output interactions and, consequently, decreases the model performance. On the other hand, the selection of non relevant variables leads to the creation of an over-dimensioned network which is more difficult to train and can show generalization problems.

The problem of variable selection is particularly important when coping with datasets concerning real world problems, when a low number of measurements of all the potential input variables is collected [8]. In these cases the number of potential inputs of the classifier is considerable with respect to the number of patterns to be exploited for model parameters tuning, i.e. in the neural network training phase.

In this paper a new method for variable selection within a classification problem is presented. The method is based on the use of a Genetic Algorithm (GA) [9] for the selection of a set of variables to be explored by a classifier which is based on the use of labeled Self Organizing Map (SOM) [10].

The SOM, although originally developed for coping with clustering tasks, has also been used with success for classification purposes [11]. This particular use of the SOM exploits its capabilities of reproducing the topology and distribution of the training data in a lower dimensional space. In this context a membership class is assigned to each reference vector of the SOM on the basis of the class which the input samples belong to.

Here a labeled SOM is exploited for using the SOM as a classifier. With respect to the standard SOM, its labeled version includes class information related to each neuron by which it was formed. This label is assigned on the basis of the class of the training data. The learning algorithm of this classifier can be divided into two steps: in the first one the standard, unsupervised training of the SOM is performed so as to generate a clusters map preserving topology and distribution of the training data. In the second step, training data are re-fed as inputs to the net together with the class to which they belong. This is done in order to assign a cluster to each corresponding class from the majority of the samples which have fired each neuron.

This approach has been widely used in different frameworks and has proved its superior efficiency with respect to other classifier-based approaches. For instance, in
II. STATE OF ART

Variable selection is a generic pre-processing step which is commonly performed within the development of models aiming at different tasks. In literature several works can be found where variable selection has been performed for function approximation [15], classification [16-18] and clustering [19] coupled to a variety of systems: neural networks [20], regression models [21] and decision trees [22]. Finally, variable selection can be a useful tool for knowledge extraction from a database, especially when developing a model related to a process or phenomenon which is not completely understood, as it highlights the variables that actually affect the investigated phenomenon.

The problem of the selection of a subset of potential input variables which are supposed to affect some output variables can be faced by means of two different approaches [23-24]:

- feature extraction,
- feature selection.

Feature extraction-based methods perform a transformation of the input space (and consequently of the original variables) in order to create a different feature set which should be more significant than the original and eventually more compact. A widely known example of feature extraction techniques is the Principal Component Analysis (PCA) [25], which performs a linear transformation of the original set of variables. The problem using a feature extraction-based approach is that the information of the original input variables is lost in the new configuration.

The feature (or variable) selection methods can be further divided into three categories: filters, wrappers and embedded methods.

Filters methods exploit a statistical test in order to perform the selection. On the basis of such tests a pertinence index is calculated for each variable. The variables which are characterized by high pertinence are finally selected. These methods are independent of the learning algorithm used to eventually train or tune the developed model. They are also computationally affordable but they can be affected by over-fitting problems. A generic scheme of a filter method is shown in Fig. 1.

An input variables subset is created through a subset generation process. The process can begin with an empty initial set or with the whole dataset and then several search strategies are applied.

The created subsets are assessed by a heuristic filter among which are informational gain [26], mutual information [27], correlation measure [28, 29], maximum relevance [30].

When the score or the optimum number of selected input variables is reached, the process finishes and the variables subset is assessed through an induction algorithm. Also the filter approaches select the best subset regardless of the classification algorithm chosen.

A simple approach consists of ranking the input variables on the basis of their relevance and performing a simple threshold comparison for the selection of the final input variables subset. In [29] the best variable subset is selected by measuring both the correlation between each variable and the class as well as the correlation among the selected variables. Finally, the relevant variables are obtained without redundancies [31]. Correlation methods such as the Pearson Correlation coefficient calculation [32] can only detect linear dependency between a variable and a target and are usually adopted for microarray data analysis [33]. Another common approach for variable selection task is the use of information theoretic criteria.

![Fig. 1. Generic scheme of filters approach](image-url)
The Mutual Information (MI) [34] between each variable and the target is calculated to obtain a measure of dependency between the density of each variable and the density of the target. Several approaches which include information theoretic criteria to the variable selection problem are proposed in literature [35-38].

Wrapper methods, which have been presented in [39], exploit the performance of the learning machine in order to select a subset of inputs according to their predictive power. Wrappers require more computational power with respect to filters but can take into account not only the relevance of the single selected variable but also its redundancy with respect to the other selected variables. Wrappers can require a high computational burden, as the learning machine is trained in order to test the various sets of candidate input variables. Figure 2 shows the flow diagram of a generic wrappers method.

In the wrapper approach [40-42] the created subsets are assessed through an induction algorithm. Wrapper approaches identify the subset of input variables which give the best classification function in terms of accuracy or of another performance indicator. In the wrapper approach the machine learning can be considered as a black box and the input variable selection can be performed also through a brute force method, although this approach is computationally complex. Actually an exhaustive approach to variable selection is not always feasible. In fact, if there are \( n \) potential inputs, there are \( 2^n \) possible subsets to test and, consequently \( 2^n \) training procedures to accomplish. When \( n \) is high, the exhaustive search becomes impractical.

A common search strategy includes the sequential algorithms, which are simple and universal [43]. This search strategy, called Greedy search strategy, can work in two directions: forward selection and backward elimination. Sequential Forward Selection (SFS) starts with an empty set of variables. These variables are progressively included into a larger subset. At each step the best candidate, in terms of accuracy of the learning machine used, is added to the selected set. This iterative procedure continues until there is no further improvement in accuracy. On the other hand backward elimination starts with all available variables and progressively deletes the least promising one. The Greedy search strategies require \( n(n+1)/2 \) trainings and, as forward selection starts with a small features set, it is less expensive than the backward elimination, if it stops early. In comparison, in filter approach there is no search and only \( n \) training procedures are required.

This approach is more expensive than the filter approach, because the induction algorithm is trained for each subset but selected subsets are more significant than those selected in the filter approach where the selection depends on features redundancy or relevance.

Embedded methods, in contrast to filter and wrapper approaches, do not separate the learning phase and the variable selection phase. They also perform variable selection during the training of the learning machine [40]. This combination is very efficient in terms of both computational cost and appropriate selection of variables. However, it requires specifically designed learning methods and systems. The advantages of embedded methods include the interaction with the classifier and a higher computational complexity with respect to wrapper methods. Feature selection is integrated in the machine learning part and is used for model generation.

There are some embedded methods which are also nested subset methods and perform forward selection or backward elimination. At each step only one feature is considered to be added or removed) requiring only \( n \) trainings. The embedded approach can reach the similar performance of the wrapper methods with the same low computational cost of the filter methods. Xiao et al. [44] proposes a novel Embedded Feature Selection Filter (EFSF) which exploits the sequential feature selection method, by incrementally adding the most important variables. This approach exploits the mass function to combine information obtained from variables in an embedded way, by reducing the computational cost from the original SFS. In Figure 3 a generic scheme of embedded methods is shown.
When treating large datasets, filter methods considering many attributes are the best approach, as wrapper methods are too expensive. On the other hand, when the dimension of dataset is moderate, wrapper or embedded methods are preferable, as they provide a higher accuracy with respect to filter methods.

Some hybrid models are also proposed in [45-46], which combine and take advantages from several approaches.

III. GA FOR THE SELECTION OF SOM INPUT VARIABLES

The basic idea that lies behind the approach proposed here for variable selection consists in exploiting the search capabilities of GA in order to find the best performing combination of input variables in terms of the accuracy of the classifier. Moreover, GAs are used in the present case not only for the selection of the variables to be used by the network but also for setting some parameters of the SOM itself. In particular, the varied parameters are as follows:

- Dimension of the net
- Topology function
- Distance function
- Steps for neighborhood to shrink to 1
- Initial neighborhood size.

During the search process the GA explores and evaluates the classification performance of different combinations of variables and network parameters and finally returns the optimal one.

Within the GAs framework, each candidate set of input variables and SOM parameters is coded into a so-called chromosome, as depicted in Figure 4.

The first part is devoted to the specification of those variables which are exploited by the SOM and is coded by means of a binary array, where each bit corresponds to a potential input variable: 1-valued bits represent a variable which has been selected, 0-valued bits correspond to non-selected variables. The second part of the chromosome specifies the SOM dimension by the means of two integer values stating the dimension of the employed two-dimensional SOM. The third part of the chromosome specifies how neurons are originally arranged in physical positions according to a topology function.

The topology function can arrange the neurons in three different ways, namely grid, hexagonal or random topology, which are coded by an integer number (1, 2 and 3, respectively). The fourth part is an integer value (which can assume three different values, i.e. 1, 2 and 3) that identifies the distance between neurons. The three available options are: Euclidean distance, link distance (a layer distance function used to find the distances between the layer's neurons) and Manhattan distance.

The GA initially creates a population of 20 chromosomes which are evenly distributed on the search space. Afterward, the GA evaluates their goodness through the fitness function. Here the fitness function provides a number $f$ in the range $[0,1]$ representing the rate of correct classifications achieved by the labeled SOM. This is trained by means of the selected variables and is dimensioned according to the dimension coded in the chromosome. More in detail, for the evaluation of a candidate solution the fitness function performs the following steps:

1. The SOM is created according to parameters specified by the candidate solution and trained by using the training dataset as input including only the selected variables. The SOM is trained on the basis of the standard algorithm presented by Kohonen in [47]. Once the training is completed, the labeling phase takes place and each neuron is associated to a target class. The class assigned to an arbitrary neuron corresponds to the class to which the majority of the patterns associated to the neuron belong.

2. The accuracy of the labeled SOM, which also represents the fitness of the GA, is calculated by simulating the classification of the observations of the training set as follows: Each pattern is associated to a neuron of the SOM according to the metric distance used during the SOM training. Afterwards the pattern is classified as belonging to the label/class to which the corresponding neuron is associated. It is also possible that a pattern is associated to a neuron whose class is not decided due, for instance, to the balance of the labels gathered during the labeling phase. In this case the result of the classification is determined on the basis of the labels associated to the neurons lying in the vicinity of the winning neuron. This is done in order to exploit the capabilities of the SOM in maintaining the distribution and topology of the original input space. A concise scheme of the operation pursued by the fitness function is depicted in Figure 5.

![Fig 4. Scheme of the chromosome structure: part of the chromosome is devoted to the coding of the variables selection, the remaining part to the SOM parameters.](image-url)
3. During the optimization, the process implements the GAs crossover and mutation in order to explore the space of the solutions and exploit the knowledge acquired during the search. The crossover operation involves two parent chromosomes and produces a son candidate solution that inherits genes from both parents. In particular, as far as the binary part (see Figure 6a) of the chromosome is concerned, in order to determine each gene of the son, the corresponding gene of both parents is considered. If the parents have the same value (0/1) for that gene, the son inherits the same value; otherwise it inherits a value randomly chosen between the ones characterizing the parents. The same procedure is applied to select the genes of the integer part of the chromosome (see Figure 6b).

The mutation operator randomly modifies a gene of the selected chromosome and switches it from 1 to 0 or vice versa in the binary part, while adds or subtracts a number in the integer part of the chromosome. In particular, concerning the net dimension, the topology function, the distance function and initial neighborhood, the mutation operator randomly adds or subtracts 1 according to the maximum or minimum allowed value. Concerning the step parameter the criterion is the same but the value 10 (instead of 1) is randomly added or subtracted.

4. The eventual verification of the stopping condition is checked. If such condition is verified, the solution with the best fitness function is returned, which corresponds to the best of the input variables and SOM parameters set, otherwise steps 1 to 3 are iterated.

As far as the stopping condition of the GAs is concerned, two criteria have been applied. The algorithm stops after the achievement of a predetermined number of iterations or when the accuracy of the classification achieved by the best performing chromosome reaches the unitary value. Figure 7 depicts a synthetic scheme of the GA-based selection of input variables for the labeled SOM-based classifier. The advantage of the proposed method lies in synergy between the GA, which is used for the selection of the variables to be used, and the classifier (in this case a labeled SOM for the classification purpose) exploiting the selected variables. The combined work of GAs and labeled SOM affects both the variables selection and the parameters of the classifier. Furthermore the use of GAs in this framework provides very fruitful results due to exploration capabilities of this method. GAs, in fact, are capable of exploring the search space in an effective manner, by limiting the number of training iterations of the labeled SOM, which represent the most time-consuming step of the whole process.

Figure 5. Flow-chart representation of the operations performed by the fitness function in order to evaluate the goodness of candidate solutions.

Figure 6a. An example of the employed crossover operation in the binary part of the chromosome.

Figure 6b. An example of the employed crossover operation in the integer part of the chromosome.

Figure 7. A flow-chart representing the GAs based variable selection system.
IV. TEST OF THE PROPOSED METHOD ON SEVERAL DATABASES.

A. Datasets description

In order to demonstrate the effectiveness of the proposed method, some tests have been performed by exploiting five datasets belonging to the UCI repository [48].

- DATABASE 1. WISCONSIN BREAST CANCER DATASET (WBCD).
This medical dataset collects information concerning the analysis of biopsies conducted on patients who are suspected of breast cancer [49]. Each sample consists of a set of 9 measures related to the analysis of the biopsy image and a classification variable assigning the benign or malignant nature of the tumor. The aim of the classifiers within this problem is to assign to each input pattern the correct diagnosis. The dataset is formed by 699 observations: 458 (65.5%) of them correspond to benign tumors while the remaining 241 (34.5%) to malignant ones.

- DATABASE 2. IRIS.
This dataset [50] (widely adopted in literature especially in the works dealing with classification procedures and methods) includes 3 classes of 50 observations corresponding to 3 types of iris plant: setosa, versicolours, verginia.

- DATABASE 3. LIVER DISORDERS (BUPA).
This medical dataset (which has been made available by BUPA Medical Research Ltd. and is therefore indicated in the following as BUPA dataset) collects information about an analysis conducted on patients with possible liver disorders. The task of this analysis is to understand if a given male patient suffers from alcoholism. The database includes 6 variables; the first 5 are related to blood tests while the last one represents the number of half-pint equivalents of alcoholic beverages that are drunk per day by the patient. The number of instance is 345 and 145 of them (42.03%) refer to alcoholic patients.

- DATABASE 4. NEW-THYROID
This medical dataset collects information related to an analysis on thyroid diseases. The aim of the analysis is to determine if a patient is healthy (class 1) or suffers from hyperthyroidism (class 2) or hypothyroidism (class 3) [51]. The dataset contains 5 features and 215 instances.

- DATABASE 5. ZOO
This is a simple dataset used to classify animals into 7 different classes on the basis on their features. The dataset contains 101 instances and 17 input Boolean variables.

Table 1 summarizes the features of the datasets that have been used for the tests.

The five datasets have been used for testing a labeled SOM classifier that has been designed by exploiting the above-described feature selection method. Each dataset has been initially divided into a training set (containing 75% of the whole observations) and a validation set (containing 25% of the whole observations) by maintaining the unaltered original distribution of the target of the starting dataset.

B. Results

Table 2 shows the obtained result in terms of accuracy, number of variables and selected SOM parameters.

The proposed approaches reaches 100% accuracy on two of the five available datasets, namely the Iris and New-thyroid datasets, and accuracy is high also on WBCD and Zoo datasets. Only on the BUPA dataset the accuracy is of 78%, which is, however, a good result, especially if compared to those achieved by other classical methods.

Table 3 shows a comparison with other classical methods, in particular Sequential Feature Selection (SFS) [43], Sequential Backward Selection (SBS) [43].

Correlation calculation (Corr) [28, 29], Mutual Information (MI) calculation [27]. Additionally, the results which are obtained by considering all the available input variables as well as through the exhaustive (or brute-force) method are shown.

Table 1  Summary of the datasets that have been exploited in the tests.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># instances</th>
<th># var.</th>
<th>CLASSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBCD</td>
<td>699</td>
<td>9</td>
<td>2 classes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1 = 34.5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C2 = 65.5%</td>
</tr>
<tr>
<td>Iris</td>
<td>150</td>
<td>4</td>
<td>3 classes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1 = 33.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C2 = 33.3%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C3 = 33.3%</td>
</tr>
<tr>
<td>BUPA</td>
<td>345</td>
<td>6</td>
<td>2 classes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1 = 42.03%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C2 = 57.97%</td>
</tr>
<tr>
<td>New-thyroid</td>
<td>215</td>
<td>5</td>
<td>3 classes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1 = 69.77%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C2 = 16.28%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C3 = 13.95%</td>
</tr>
<tr>
<td>Zoo</td>
<td>101</td>
<td>16</td>
<td>7 classes:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C1 = 40.59%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C2 = 19.80%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C3 = 4.95%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C4 = 12.87%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C5 = 3.96%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C6 = 7.92%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>C7 = 9.901%</td>
</tr>
</tbody>
</table>

Table 2 Summary of the tests results.

<table>
<thead>
<tr>
<th>Dataset</th>
<th># var</th>
<th>Net dim</th>
<th>Topol.</th>
<th>Dist</th>
<th>Steps</th>
<th>In. Neigh</th>
<th>Acc %</th>
</tr>
</thead>
<tbody>
<tr>
<td>WBCD</td>
<td>5</td>
<td>7x3</td>
<td>2</td>
<td>2</td>
<td>20</td>
<td>4</td>
<td>99.3</td>
</tr>
<tr>
<td>Iris</td>
<td>3</td>
<td>6x2</td>
<td>1</td>
<td>3</td>
<td>70</td>
<td>2</td>
<td>100</td>
</tr>
<tr>
<td>BUPA</td>
<td>4</td>
<td>7x3</td>
<td>2</td>
<td>1</td>
<td>90</td>
<td>2</td>
<td>78</td>
</tr>
<tr>
<td>New-thyroid</td>
<td>4</td>
<td>4x1</td>
<td>2</td>
<td>3</td>
<td>62</td>
<td>1</td>
<td>100</td>
</tr>
<tr>
<td>Zoo</td>
<td>2</td>
<td>4x4</td>
<td>1</td>
<td>3</td>
<td>70</td>
<td>2</td>
<td>96</td>
</tr>
</tbody>
</table>
Finally the proposed method contributes not only to the classification method but also to the achievement of a deeper comprehension of the phenomenon under consideration. This is because it highlights the variables which are more involved in the considered problem especially when relationships between input variables and classification results are very complex, namely in the cases where the application of AI-based classifiers (such as the labeled SOM considered here) is mostly advisable.

V. CONCLUSION

In this paper, the problem of the selection of the variables to be used as inputs to a labeled SOM for a classification task has been faced by means of GAs. Through the proposed GA-based optimization procedure, it is possible to determine not only the combination of input variables which maximizes the classification accuracy of the classifier but also the optimal parameters set of the SOM. This method has been tested on several databases and proved its efficiency. In a generic problem, where a database coming from a real world classification task is available, but the knowledge of the considered phenomenon is quite poor, the set of variables which are selected through the proposed method can be interpreted by experts on the process or phenomenon under consideration in order to get a deeper insight into the way each variable affects the final classification.

REFERENCES
