SUPERVISED METHODS FOR FEATURE EXTRACTION

Oswaldo Ludwig Júnior [1], A. C. de Castro Lima[1], Leizer Schnitman [1], J.A.M.Felippe de Souza[2]

[1] Universidade Federal da Bahia
Department of Electrical Engineering
Salvador-BA-Brazil, e-mail: oludwig@terra.com.br

[2] Universidade da Beira Interior
Department of Electrical Engineering
Covilhã - Portugal, e-mail: felippe@dem.ubi.pt

ABSTRACT

It is present in this paper the feature extraction for pattern recognition tasks. It is proposed two approaches. In the first, it is used weights to scale the coordinates of the features vector in order to increase the precision of statistical classifiers. Genetic algorithm is intended to do weight adjustments. In the second approach the Battacharyya metric is suggested. Theses approaches make possible the feature vector compression by the elimination of coordinates non-pertinent to the classification problem in subject. There are other techniques like Principal Components Analysis (PCA) or entropy, but these approaches do not consider the target output. This fact implicate in the impossibility of non-pertinent features identification.

Keywords: genetic algorithm, pattern recognition, statistics, artificial intelligence.

1. INTRODUCTION

There are several approaches to the problem of Patterns Recognition (PR): the statistical approach [1], fuzzy [8], connectionist and PR based on knowledge [9]. All of them require an intelligent method for feature extraction.

The classification problem is subdivided into two different tasks: the features extraction and the comparison of these features with perfect models (i.e. noise free and representative of their respective patterns). These tasks are usually accomplished by two modules, which are denominated as feature extractor and classifier [18, 19, 20].

The extracted features are commonly composed by a set of numeric values that should be enough for the appropriate representation of the input data, with respect to the classification task in subject. The feature vector represents this set of values, so that a point in a features space can represent an object.

There are two different kinds of approaches to the feature extraction task: those based on the knowledge of a specialist and those that apply supervised automated methods. This paper deals with the second kind of approach.

This work presents the use of genetic algorithm (GA) and the Battacharyya metric to feature extraction tasks. However, there are other techniques to feature extraction like Principal Components Analysis (PCA) [6] or Entropy Analyses [1], but both approaches do not consider the target output (i.e. non-supervised methods). This fact implicate in the impossibility of non-pertinent feature identification. No pertinent features acts as noise that decreases the classifier performance.

The recognition of fail patterns has direct application to industrial automation processes. In this context, the design of the features extractor is critical to the appropriate classifier performance.

Section 2 presents the Battacharyya metric. Sections 3 and 4 describe the use of genetic algorithm (GA) in the weights adjustment including an example. Finally, the Conclusions are presented in Section 5.

2. THE BATTACHARYYA METRIC

The Battacharyya distance [14] is a measure of removal between two probabilities distributions. These distributions are characterized by their respective probability density functions. Lets $f(x_n|c_1)$ and $f(x_n|c_2)$ be the probability density functions of the feature $x_n \in R$, associated to classes c_1 and c_2 respectively. The Battacharyya distance among these classes is defined by:

$$B_n = \frac{1}{\log(\rho_n)} \qquad (2.1)$$

where:

$$\rho_n = \int_R \sqrt{f(x_n|c_1) \cdot f(x_n|c_2)} \cdot dx_n \qquad (2.2)$$

Figure 1 illustrates a case of $\rho \cong 0$. This implicates in a biggest *B* distance. In other hand, Figure 2 shows a case of a larger ρ, which leads to a smaller *B* distance.

The main idea of Battacharyya's method is the selection of the features that have the biggest *B* distance among the classes. These features have more capacity to separate classes.

The use of the Battacharyya´s distance implicates in the probability density functions determination. These functions must be calculated to each one of the *n* coordinates of the characteristics vector in respect to each one of the *m* classes. Note that a total of $n \cdot m$ functions must be estimated.

This work considers uncorrelated features. That simplification result:

$$P\{x = \chi\} = P\{x_1 = \chi_1\} \cdot P\{x_2 = \chi_2\} \cdot \ldots \cdot P\{x_n = \chi_n\}$$

Usually, the first step in the estimate of a *pdf* is associate this *pdf* to a family of functions. As an example one can mention Gaussians functions. The second step is the parameters adjustment.

This process has an elevated computational cost and implicates in a great set of examples. An alternative approach is the features discretization. In this approach the Equation 2.2 becomes:

$$\rho_n = \sum_{x \in \Omega} \sqrt{P\{x_n = \chi|c_1\} \cdot P\{x_n = \chi|c_2\}} \qquad (2.3)$$

In case of discrete values a normalized histogram is enough to estimate *B*.

If the classification problem has *m* different classes, the Equation 2.3 becomes:

$$\rho_n = \sum_{x \in \Omega} \sqrt[m]{P\{x_n = \chi|c_1\} P\{x_n = \chi|c_2\} \ldots P\{x_n = \chi|c_m\}} \qquad (2.4)$$

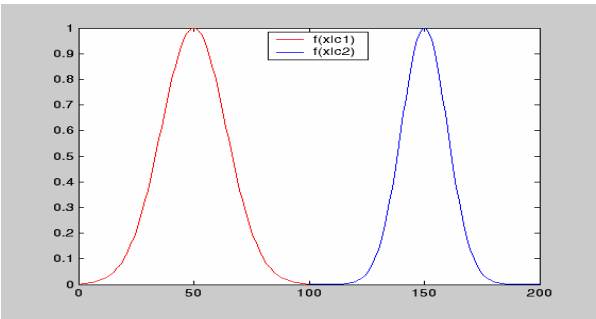

Figure 1. *pdf* of *x* associated to the classes c_1 and c_2 with great *B* distance.

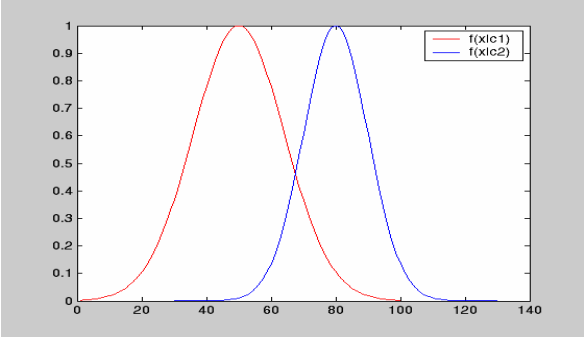

Figure 2. *pdf* of *x* associated to the classes c_1 and c_2 with smaller *B* distance.

3. STATISTICAL CLASSIFIER SCALED BY WEIGHTS

The statistical classifier performance can be optimized by the weights application on the normalized coordinates of the feature vector. This approach leads to alterations in the relative scales between the dimensions or axes of the feature space. Therefore, more relevant $\boldsymbol{x_j}$ are scaled by largest weights $\boldsymbol{k_j}$. This provide a largest influence in the computation of the Euclidean distance between an entrance data $x=[x_1, x_2,\ldots, x_j]^T$ and a prototype $\mu=[\mu_1, \mu_2,\ldots, \mu_j]^T$, whose expression for this classifier is:

$$\tilde{r} = \sqrt{\sum_J (k_j (x_j - \mu_j))^2} \qquad (3.1)$$

where $j=1, 2,\ldots J$ is the coordinate index of the features vector.

Notice that correlate or irrelevant features can be scaled by smaller weights (i.e. present smaller participation in the Equation 3.1).

After the coordinates normalization of the feature vector, the prototype μ_i representative of a class or pattern *i* is computed.

3.1. Common Limitations to the Statistical Approach

The determination of relevant properties that will compose the feature vector, usually demands a good understanding of the application problem, once these features are strongly dependent on each particular problem. Thus, to the same set of data, one can extract different feature vectors based on the type of classification task required. As an example, one can mention *recognition of phonemes* and *recognition of announcer* tasks. Both receive the same set of data; however, the features vectors adopted should be different. It can be consider that the feature extraction through a specialist is more an art than a science.

An inadequate choice of the features vector can decrease the classifier performance. Some common problems that occur due to an inadequate feature extraction can be mentioned:

- Correlated features: in this case the information contained in the vector of features may be redundant, because some of their coordinates can be approximated by a linear combination of the other ones. Thus, the amount of information contained in this vector may be insufficient to data appropriated characterization.
- Features in inadequate scales: feature relevance may mask the solution. Moreover, numerical problems may occur.
- Feature vectors that will require non-linear decision surfaces for the pattern classification. It means that a more complex classifier (i.e. higher V-C dimension) will be also required.
- Inadequate features: the features are, simply, inadequate or insufficient and they are not able to pattern differentiation.

These problems can be avoided by the initial selection of a great number of features. After the first training stage, some of these features can be discarded in function of the adjusted weights values k. In a second training stage, the classifier is adjusted again for the new features vector (i.e. smaller dimension vector). In this stage the training can be more exhausting, due to the smallest number of adjusted parameters.

It is important to notice that, like the other classifiers based on the smallest Euclidean distance, this classifier possesses linear decision surfaces. Thus, this classifier is not capable to treat non-lineally separable patterns. The traditional problem XOR [6] exemplifies this limitation. The inspection of the Figure 3 reveals that the change of axes scales (i.e. introduction of the weights) does not avoid the prototypes μ_1 and μ_2 coincidence.

Quadratic decision surfaces as proposed in the classifiers based on the Mahalanobis distance are more capable than minimum Euclidean distance classifiers for patterns separation. However, the computational cost of the covariance inverse matrix increases proportionally to the square of the features vector dimension. Therefore, it becomes unviable for many practical applications.

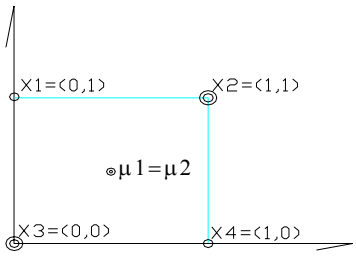

Figure 3. Patterns separation of the XOR problem.

4. GA APPLIED TO WEIGHTS ADJUSTMENTS

The classifier weights adjustments can be made with the use of genetic algorithm (GA). A fitness function must be defined. This implicates in the adoption of a performance evaluation method.

The classifier performance is a function of the success tax on the database evaluation. The algorithm should be evaluated with unknown data. The database is usually divided into two groups, one for training and other for performance evaluation $\aleph_a$. The classifier should predict the class of each one of the test examples. The results are compared with the target values (i.e. correct class). The success tax ***a*** is equal to the total number of correct predictions ***c*** divided by the total number of predictions:

$$a = \frac{c}{|\aleph_a|} \tag{4.1}$$

The crossed validation method guarantees larger precision in the performance evaluation of a classifier. This method divides the database in ***k*** parts. One of these parts is used to test and the other ***k***-1 parts are used in the training process. The performance is the average on all the tests (i.e. over all of the parts ***k***).

The ***a*** parameter is used to compose the fitness function. Its fitness ***f*** and its chromosome compose each individual. A vector $C \in \Re^j$ containing ***j*** weights, corresponding to ***j*** features, characterizes the chromosome.

Correlate features, features that contain low information level (i.e. low entropy) or no pertinent information acts as noise that decreases the classifier performance.

Figure 4 illustrates the features scaled by weights application. Notice that before the weights application the ***x*** data belongs to $\boldsymbol{\mu_2}$. Supposing that the horizontal feature contain low entropy or no pertinent information, GA may scale the horizontal feature by $\boldsymbol{w_1}$=0.5 and the vertical feature by $\boldsymbol{w_2}$=1.0. In this case, the classifier performance will change and the ***x*** data will belong to $\boldsymbol{\mu_1}$.

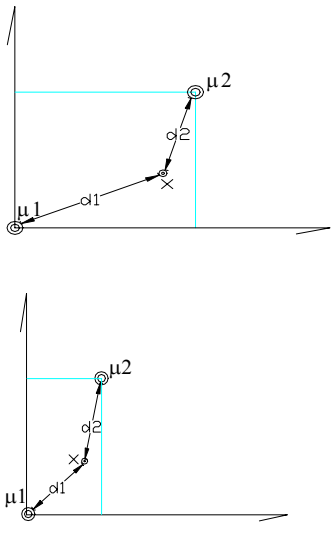

Figure 4. Features scaled by weights application

GA should find a relationship among the weights in order to maximize the classifier efficiency, in other words, the parameter $\boldsymbol{a}$. However, in case of totally correlate features or features that always assume a single value (i.e. with null entropy) GA may not find smaller weights for these coordinates. This fact occurs due to the absence of influence of the foregoing coordinates in the classifier performance. Such features are redundant or useless.

The foregoing features is undesirable, due the fact of demanding a larger computational effort, besides decreases on the understanding of the problem by a human supervisor in case of high dimension of the features vector. It is necessary that the GA find smaller values to the weights that scale irrelevant features, although these do not decrease the classifier efficiency. That is solved by the adoption of the fitness function:

$$f = c_1 \cdot s(k) + c_2 \cdot a \tag{4.2}$$

where $\boldsymbol{c_1}$ and $\boldsymbol{c_2}$ are constants defined by the planner and *s(.)* is the *standard deviation* statistical function over the weights vector *k*, defined by the equation:

$$s(k) = \sqrt{\frac{\sum_{j=1}^{J} (k_j - \bar{k})^2}{J}} \tag{4.3}$$

With this fitness function, GA is driven to minimize the weights that scale redundant or irrelevant features in order to maximize the standard deviation *s(k)*.

After a first training stage, the coordinates scaled by lowest weights may be eliminated.

It is possible the adoption of the crossed validation method on the GA applications by the random selection of a new test part (i.e. subset of examples to the determination of the individual fitness) to each generation. To each generation, the prototypes of each class are calculated by the application of the Equation 2.1 on the k-1 parts that is not being adopted for determination of the individual fitness.

More details on GA are found in [1] and [2].

A simple example was implemented in the MATLAB to illustrate the proposed method (see Figure 5). In this example, the classifier should relate 100 points lineally separable to 2 patterns (i.e. red and blue). The patterns were generated around the $\boldsymbol{\mu_{red}}$=[4,5 4,5] e $\boldsymbol{\mu_{blue}}$=[4,2 5,5] prototypes, with standard deviation $\boldsymbol{s}$=1 in an uniform distribution. GA should find the weights vector $\boldsymbol{k}$=[$\boldsymbol{k_1}$ $\boldsymbol{k_2}$]. This vector must adjust the axes scales of the features space.

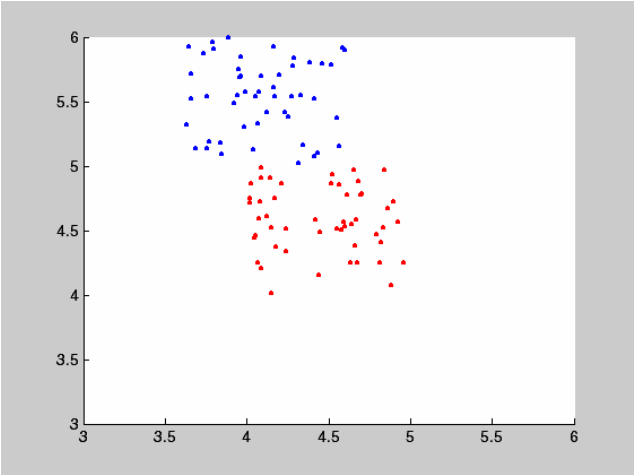

Figure 5. Classification of 2 lineally separable patterns

Through a visual analysis, it is possible to verify that the vertical coordinate is more relevant than the horizontal coordinate to the pattern characterization. GA must adjust $\boldsymbol{k_1}$<$\boldsymbol{k_2}$.

The features below describe briefly the GA applied in this paper:

- Initial population = final population =50 individuals.
- Chromosomes: weights vector $c \in \Re^2$ on decimal base with bounded values [0, 1].
- Selection criteria:

$$p_i = \frac{f_i}{\sum_{i=1}^{50} f_i} \tag{4.4}$$

Mutation Genetic Operator: only one individual has the genes of its chromosome inverted.

- Crossover Genetic Operator BLX-α defined by:

$$c[n+1] = \alpha\, c_1[n] + (1-\alpha)\, c_2[n] \tag{4.5}$$

where *n* is the generation, c_1 and c_2 are parents of *c* and $\alpha \in [0, 1]$, with normal density probability.

- Elitism Genetic Operator: only the most capable individual.
- Fitness function: $f = 0{,}9s(k) + a$ (4.6)
- Convergence criteria: after 50 generations.

The individual fitness converges quickly and the individual selection probability, based on Equation 4.4, becomes approximately the same for all individuals (i.e. a random optimization process. To solve this problem, it was decided to use a corrected fitness function fc_i:

$$fc_i = f_i - 0.8 \cdot \min_i(f_i)\ , \quad i = 1,2,...50 \tag{4.7}$$

After the training section, the GA adjusted the vector of weights $\boldsymbol{k}$=[0,13 0,94], conform was already waited (i.e. $\boldsymbol{k_1}$<$\boldsymbol{k_2}$).

In case of a great features vector dimension, the convergence may be a hard process. A "seeding" method introduces in the initial population, some solutions found by the use of other methods. In this case, solutions may be obtained by the application of entropy and correlation methods.

For example, one can consider the application of entropy methods. If considered that features are represented by a constraint set of discrete values, it is possible to associate each feature to a discrete random variable.

Computation of the entropy of a discrete random variable $\boldsymbol{X}$ requires the computation of the amount of information $\boldsymbol{I}$ revealed after occurrence of the event $\boldsymbol{X}=\boldsymbol{x_i}$. Where $\boldsymbol{I}$ is related to the $\boldsymbol{x_i}$ occurrence rarity. It means that observation of expected events brings low information level. On the other hand, a rare event is surrounded by a very specifics circumstance, which brings new information.

Consider probability is $\boldsymbol{p_i}$ as the probability of the event $\boldsymbol{X}=\boldsymbol{x_i}$ occurrence, then the amount of information is defined as:

$$I(x_i) = \log(\frac{1}{p_i}) = -\log(p_i) \qquad (4.8)$$

Notice that the inverse relationship with the probability denotes the notion of rarity. Since the scale is logarithmic, if $\boldsymbol{p_i}=1$ then $\boldsymbol{I}(\boldsymbol{x_i})=0$ and means that events that are 100% predicable does not contain any new information.

Considering $\boldsymbol{N}$ possible values of $\boldsymbol{x_i}$ that $\boldsymbol{X}$ can assume, the entropy is computed as:

$$H(X) = E\left[I(x_i)\right] = \sum_{i=1}^{N} -\log(p_i)\cdot p_i \qquad (4.9)$$

where E[.] is the expectation statistical operator.

Each feature may be scaled by its entropy (i.e. $\boldsymbol{k_j}=H(\boldsymbol{x_j})$), compounding an chromosome.

5. CONCLUSIONS

There are limitations in the use of statistical classifiers (i.e. inability to treat non-lineally separable patterns). However, the evolutionary methodology used in the weights estimate makes possible the relevant features extraction. In case of problems that demand more complex decision surfaces, more efficient classifiers can process features. RNAs feedforward multilayer can be used [13].

In case of a larger number of features, the Battacharyya distance it is more efficient than GA methods. A great search space hinders the convergence. In other hand, the GA approach maximizes the real performance of the classifier. Therefore, this technique considers all the indicative of feature relevance, such as covariance, entropy and pertinence.

ACKNOWLEDGMENTS:

This work has the support of FAPESB (Brazil) and FCT (Portugal).

REFERENCES

[1] Yasunaga, M.; Nakamura, T.; Yoshimara, I and Kim, J.H.: "*The kernel-based pattern recognition system designed by genetic algorithm*", *IEICE Trans. on Information and Systems*, v E84, n 11, p 1528-1539, Nov 2001.

[2] L. Eshelman, D. Shaffer, "Real-coded genetic algorithms and interval-schemata", Foundation of Genetic Algorithms 3. San Mateo, CA, 1992.

[3] Lippmann, R.P., "*Pattern classification using neural networks*", IEEE Communications Magazine, vol.27, 1989.

[4] Kohonen, T.: "*Exploration of very large databases by self-organizing maps*", International Conference on Neural Networks, vol I, 1997.

[5] Haykin, S. et al.: "*Classification of radar clutter in air traffic control environment*", Proceedings of the IEEE, vol.79, 1991.

[6] Haykin, S.: "*Redes Neurais, princípios e práticas*", Bookman

[7] Parbhane RV, Tambe SS, Kulkarni BD., "*ANN modelling of DNA sequences: new strategies using DNA shape code*", Comput Chem 2000; 24: 699-711.

[8] S.N. Srihari, "*High-Performance Reading Machines"*, *Proceedings of the IEEE*, 80(7), July 1992, 1120-1132.

[9] C. C. Yang, M. M. Marefat, and R. L. Kashyap, "*Automated Visual Inspecion Based on CAD Models*", Proceedings of IEEE International Conference on Robotics and Automation, San Diego, CA, May 8-13, 1994.

[10] Morgan Kaufmann, "*Information Visualization in Data Mining and Knowledge Discovery*" , 2001

[11] Carreira-Perpinan, M. A.: "*Compression neural networks for feature extraction: Application to human recognition from ear images*", MSc thesis, Faculty of Informatics, Technical University of Madrid, Spain, 1995.

[12] Kim, S.S.; Lee, D.J.; Kwak, K.C.; Park, J.H. and Ryu, J.W.: "*Speech recognition using integra-normalizer and neuro-fuzzy method*", *IEEE Conf. on Signals, Systems and Computers*, v 2, p 1498-1501, 2000.

[13] Ludwig, O., Schnitman L., Souza J.A.M.F., Lepikson H., "*A comparative analysis between conventional approaches and connectionist methods in pattern recognition tasks*", Proceedings of the IASTED International Conference on Artifitial Intelligence and Applications, pp 639-644, Innsbruck, Austria, Feb 2004.

[14] Kailath T., "The Divergence and the Battacharyya Distance Measures in Signal Selection", IEEE Trans. Commun. Theory, COM 5, pp.52-60, 1967.