ELSEVIER

Pattern Recognition Letters 16 (1995) 1069–1078

Pattern Recognition Letters

An evolutionary approach to training relaxation labeling processes

Marcello Pelillo [a,*], Fabio Abbattista [b], Angelo Maffione [b]

[a] *Dipartimento di Matematica Applicata e Informatica, Università di Venezia, Via Torino 155, 30173 Venezia Mestre, Italy*

[b] *Dipartimento di Informatica, Università di Bari, Via E. Orabona 4, 70126 Bari, Italy*

Received 2 January 1995; revised 4 April 1995

Abstract

In a recent work, a learning procedure for relaxation labeling algorithms has been introduced which involves minimizing a certain cost function with classical gradient methods. Although the results obtained so far are extremely promising, the gradient-based learning algorithm suffers from some inherent drawbacks that could prevent its application to real-world problems of practical interest. Essentially, these include the inability to escape from local minima and its computational complexity. In this paper, we propose using genetic algorithms to solve the relaxation labeling learning problem in an attempt to overcome the difficulties with the gradient algorithm. Experiments are presented which demonstrate the superiority of the proposed approach both in terms of quality of solutions and robustness.

Keywords: Learning; Relaxation labeling; Genetic algorithms; Line labeling; Part-of-speech disambiguation

1. Introduction

Relaxation labeling processes are a broad class of popular techniques within the pattern recognition and machine vision domains (Rosenfeld et al., 1976; Hummel and Zucker, 1983; Davis and Rosenfeld, 1981). They are parallel iterative procedures that attempt to combine local and contextual information in order to remove, or at least reduce, labeling ambiguities in classification problems where local measurements may be noisy or unreliable. In (continuous) relaxation labeling models, contextual information is embedded in a set of real-valued *compatibility coefficients*, which quantitatively express the degree of agreement of label configurations. In the past, a number of authors have stressed the importance of deriving ''good'' compatibility coefficients, and several heuristic statistical-based interpretations such as correlation (Rosenfeld et al., 1976) or mutual information (Peleg and Rosenfeld, 1978) have been formulated.

Recently, a novel approach for determining the compatibility model of relaxation labeling procedures has been introduced which views the problem as one of *learning* (Pelillo and Refice, 1994) – this has been shown to clearly outperform the standard statistical approach. According to this standpoint, compatibility coefficients are derived in such a way as to optimize the performance of the relaxation labeling algorithm over a sample of training data. This amounts to minimizing a certain cost function

* Corresponding author. Email: pelillo@moo.dsi.unive.it

which quantifies the degree of "goodness" of a given set of compatibility strengths, so that the learning task is formulated in terms of an optimization problem. This is essentially the approach that has recently become so popular within the neural network community (see (Pelillo and Refice, 1994) for a discussion concerning the relations between neural network and relaxation labeling learning algorithms).

In (Pelillo and Refice, 1994), classical gradient techniques were used to solve the relaxation labeling training problem. However, gradient-based learning procedures exhibit some inherent limitations that could prevent them from being applied to high-dimensional problems of practical interest. To begin with, the surface defined by the cost function can be a very complex one and can have many poor local minima, so that the gradient algorithm may be easily trapped into such a solution. Another problem with the gradient procedure concerns its computational complexity as it requires in each step a number of operations of the order of the fourth power of the number of labels (or classes) of the problem at hand. This makes the algorithm excessively expensive in problems where a large number of classes is involved (for example, in character or text recognition the number of character categories may range from a few tens to some thousands). As a final remark, we note that some relaxation schemes are even non-differentiable (see, e.g., Zucker et al., 1981) and this completely prevents the gradient algorithm from being applied.

In this paper, we attempt to overcome the limitations of gradient-based relaxation labeling learning procedures by proposing the use of genetic algorithms (GAs) (Goldberg, 1989; Holland, 1992) which have recently gained wide popularity as optimization procedures, especially for their properties of robustness and simplicity. Genetic algorithms exhibit several advantages over gradient-based optimization procedures. Firstly, they efficiently explore complex search spaces and are able to find globally near-optimal solutions without being trapped into local optima. Also, they involve simple arithmetic operations and do not require any auxiliary information about the objective function (like derivatives) other than the values of the function themselves. In addition, we found GAs attractive for our learning problem because they are computationally less expensive than gradient methods, requiring in each step a time roughly proportional to the square of the number of labels (although more training cycles are typically needed to find a solution). Finally, since GAs work on populations, they lend themselves well to parallel implementation (Goldberg, 1989).

The paper is organized as follows. In Section 2, we briefly introduce relaxation labeling processes and formulate the learning problem as one of optimization. In Section 3, we describe GAs and the way in which they are applied to the present learning task. Section 4 presents some experimental results and, finally, Section 5 concludes the paper.

2. Relaxation labeling and the learning problem

Relaxation labeling processes involve a set of objects $\boldsymbol{B} = \{b_1, \dots, b_n\}$ and a set of labels $\boldsymbol{\Lambda} = \{1, \dots, m\}$. The purpose is to label each object of $\boldsymbol{B}$ with one label of $\boldsymbol{\Lambda}$. By means of some local measurement it is generally possible to construct, for each object b_i, a vector $\boldsymbol{p}_i^{(0)} = (p_{i1}^{(0)}, \dots, p_{im}^{(0)})^{\mathrm{T}}$ such that $p_{i\lambda}^{(0)} \geqslant 0$ $(i = 1, \dots, n$ and $\lambda = 1, \dots, m)$ and $\sum_\lambda p_{i\lambda}^{(0)} = 1$ $(i = 1, \dots, n)$. Each $\boldsymbol{p}_i^{(0)}$ can thus be interpreted as the *a priori* (non-contextual) probability distribution of labels for b_i. By simply concatenating $\boldsymbol{p}_1^{(0)}, \dots, \boldsymbol{p}_n^{(0)}$ we obtain an initial weighted labeling assignment for the objects of $\boldsymbol{B}$ that will be denoted by $\boldsymbol{p}^{(0)} \in \mathbb{R}^{nm}$. The compatibility model is represented by a four-dimensional matrix of real-valued nonnegative compatibility coefficients $R = \{r_{ij}(\lambda, \mu)\}_{ij\lambda\mu}$; the element $r_{ij}(\lambda, \mu)$ measures the strength of compatibility between the hypotheses "b_i has label λ" and "b_j has label μ". High values correspond to compatibility and low values correspond to incompatibility. We will find it convenient to "linearize" the matrix R and consider it as a column vector $\boldsymbol{r}$.

The relaxation labeling algorithm accepts as input the initial labeling assignment $\boldsymbol{p}^{(0)} = (\boldsymbol{p}_1^{(0)^{\mathrm{T}}}, \dots, \boldsymbol{p}_n^{(0)^{\mathrm{T}}})^{\mathrm{T}}$ and updates it iteratively, taking into account the compatibility model, in order to eventually achieve global consistency. At the tth iteration ($t = 0, 1, 2, \dots$) the labeling is updated according to the following classical formula (Rosenfeld et al., 1976):

$$p_{i\lambda}^{(t+1)} = p_{i\lambda}^{(t)} q_{i\lambda}^{(t)} \Big/ \sum_{\mu=1}^{m} p_{i\mu}^{(t)} q_{i\mu}^{(t)} \qquad (1)$$

where the denominator is simply a normalization factor, and

$$q_{i\lambda}^{(t)} = \sum_{j=1}^{n} \sum_{\mu=1}^{m} r_{ij}(\lambda, \mu) p_{j\mu}^{(t)} \tag{2}$$

represents a "contribution" function that measures the strength of support that context gives to λ for being the correct label for b_i. The process is stopped when some termination condition is satisfied (e.g., when the distance between two successive labelings becomes negligible or, more commonly, after a fixed number of iterations) and the final labeling is usually used to label the objects of $\boldsymbol{B}$ according to a maxima selection criterion (Zucker et al., 1981). The preceding formulas are those originally proposed by Rosenfeld et al. (1976) which, despite their completely heuristic derivation, have been recently shown to possess interesting dynamical properties (Pelillo, 1994). Although in this work we have restricted ourselves primarily to this relaxation scheme, we would like to emphasize that the learning approach we are proposing here does not depend in any way on the particular updating rule employed, and *does* work even with non-differentiable relaxation schemes (Zucker et al., 1981).

Now, let us focus on the learning problem. Let a set of learning samples

$$L = \{L_1, \ldots, L_N\}$$

be given, where each sample L_γ ($\gamma = 1, \ldots, N$) is a set of labeled objects of the form

$$L_\gamma = \{(b_i^\gamma, \lambda_i^\gamma) : 1 \leqslant i \leqslant n_\gamma,\ b_i^\gamma \in \boldsymbol{B},\ \lambda_i^\gamma \in \Lambda\}.$$

For each $\gamma = 1, \ldots, N$, let $\boldsymbol{p}^{(L_\gamma)} \in \mathbb{R}^{n_\gamma m}$ denote the unambiguous labeling assignment for the objects of L_γ, i.e.,

$$p_{i\alpha}^{(L_\gamma)} = \begin{cases} 0, & \text{if } \alpha \neq \lambda_i^\gamma, \\ 1, & \text{if } \alpha = \lambda_i^\gamma \end{cases}.$$

Also, suppose that we have some mechanism for constructing an initial labeling $\boldsymbol{p}^{(I_\gamma)}$ on the basis of the objects in L_γ, and let $\boldsymbol{p}^{(F_\gamma)}$ denote the labeling produced by the relaxation algorithm, according to any stopping criterion, when $\boldsymbol{p}^{(I_\gamma)}$ is given as input. In general, a relaxation labeling process is a function that, given as input a vector of compatibilities $\boldsymbol{r}$ and an initial labeling $\boldsymbol{p}^{(I)}$, produces iteratively the final labeling $\boldsymbol{p}^{(F)}$. Here, we will consider the relaxation operator as a function of the compatibility coefficients only, the initial labeling being regarded as a constant.

Broadly speaking, the learning problem for a relaxation labeling process is to determine a vector of compatibilities $\boldsymbol{r}$ so that the final labeling $\boldsymbol{p}^{(F_\gamma)}$ be as close as possible to the desired labeling $\boldsymbol{p}^{(L_\gamma)}$, for each $\gamma = 1, \ldots, N$. To do this, we can define a cost function measuring the loss incurred when $\boldsymbol{p}^{(F_\gamma)}$ is obtained instead of $\boldsymbol{p}^{(L_\gamma)}$, and attempt to minimize it with respect to $\boldsymbol{r}$. As both $\boldsymbol{p}^{(F_\gamma)}$ and $\boldsymbol{p}^{(L_\gamma)}$ are composed of n_γ probability vectors (the $\boldsymbol{p}_i^{(F_\gamma)}$'s and the $\boldsymbol{p}_i^{(L_\gamma)}$'s, respectively), it seems natural to make use of some divergence measure between probability distributions. The best known of such measures is certainly Kullback's (1959) directed divergence which was in fact used in the previous work with the gradient algorithm (Pelillo and Refice, 1994), and yielded better generalization results than the more traditional quadratic error function. However, Kullback's divergence measure requires that the two probability distributions be absolutely continuous. In our case, this means that $\boldsymbol{p}_i^{(F_\gamma)} \cdot \boldsymbol{p}_i^{(L_\gamma)} \neq 0$ for all $i = 1, \ldots, n_\gamma$, which, in words, amounts to requiring that the relaxation algorithm *does* assign nonzero probability to the correct labels. Therefore, the use of Kullback's measure could cause a run-time error when, for some i, this condition is not met, and we mention that this actually occurred in our preliminary experiments with the GA. Fortunately, Lin (1991) has recently proposed a more robust information-theoretic divergence measure which does not require the absolute continuity property and is closely related to Kullback's measure. According to Lin's measure, the cost (or error) for sample γ turns out to be:

$$E_\gamma = n_\gamma - \sum_{i=1}^{n_\gamma} \log_2\left(1 + \boldsymbol{p}_i^{(F_\gamma)} \cdot \boldsymbol{p}_i^{(L_\gamma)}\right) \tag{3}$$

where "$\cdot$" denotes the inner product operator. Note that $E_\gamma = 0$ if and only if $\boldsymbol{p}^{(L_\gamma)} = \boldsymbol{p}^{(F_\gamma)}$ and it attains its maximum value when relaxation assigns null probabilities to all the correct labels. The error achieved over the entire learning set can thus be defined as

$$E = \sum_{\gamma=1}^{N} E_\gamma. \tag{4}$$

In conclusion, the learning problem for relaxation labeling can be stated as the problem of minimizing the function E with respect to $\boldsymbol{r}$. In (Pelillo and Refice, 1994), this problem is solved by means of a gradient method which begins with an initial point $\boldsymbol{r}_0$ and iteratively produces a sequence $\{\boldsymbol{r}_k\}$ as follows:

$$\boldsymbol{r}_{k+1} = \boldsymbol{r}_k - \alpha_k \boldsymbol{u}_k, \tag{5}$$

where $\boldsymbol{u}_k$ is a direction vector determined from the gradient of E, and α_k is a suitable step size. It is readily seen that the number of derivatives that are to be computed is of order of m^3 (m being the number of label values), and for each of them about m calculations are needed (Pelillo and Refice, 1994). Therefore, the overall computational complexity of the algorithm turns out to be $O(m^4)$, and this makes it unfeasible for problems where the number of labels is large.

3. Learning compatibility coefficients with genetic algorithms

Genetic algorithms are parallel search procedures largely inspired from the mechanisms of evolution in natural systems (Goldberg, 1989; Holland, 1992). In contrast to more traditional optimization techniques, GAs work with a constant-size population of points which, in GA terminology, are called chromosomes or individuals. Every chromosome is associated with a ''fitness'' value that determines its probability of surviving in the next generation; the higher the fitness, the higher the probability of survival. Clearly, in an optimization problem a chromosome's fitness must be somehow related to the corresponding value of the objective function. In the present application, each chromosome represents a vector of compatibility coefficients $\boldsymbol{r}$; each coefficient $r_{ij}(\lambda, \mu)$ is mapped onto a fixed-length string of bits, and the whole chromosome is then obtained by concatenating these strings.

The GA starts out with an initial population of S members generally chosen at random and, in its simplest version, makes use of three basic operators: reproduction, crossover, and mutation. The most popular way of implementing reproduction, commonly referred to as *roulette-wheel* selection (Goldberg, 1989), consists of choosing the chromosomes to be copied in the next generation according to a probability proportional to their fitness. Specifically, the probability that chromosome i will be reproduced is given by

$$P_i = F_i / \sum_{k=1}^{S} F_k \tag{6}$$

where F_k represents the fitness value of the kth chromosome of the current population. One problem with this mechanism is that the best individuals do not necessarily survive in future generations and this can slow down the convergence of the algorithm. Indeed, a recent theoretical study (Rudolph, 1994) has revealed that roulette-wheel selection does not guarantee convergence to the global optimum, even in infinite time. To overcome this drawback, we made use of an *elitist* reproduction mechanism (Grefenstette, 1986; Davis, 1991), which consists of copying deterministically the best individual of each generation into the succeeding one, the other members being copied according to the usual roulette-wheel strategy. This straightforward modification has been proven to guarantee asymptotic convergence to the global optimum (Rudolph, 1994; Eiben et al., 1991). After reproduction, the crossover operator is applied between pairs of selected individuals in order to produce new offspring individuals. The operator proceeds in two steps. First, two members are chosen at random; next, a cut point is determined (again) randomly and the corresponding right-hand segments are swapped. The frequency with which crossover is applied is controlled by a parameter P_c. Recall that in our application each chromosome represents a vector of compatibility coefficients; hence, the crossover point is allowed to fall only at the boundary between two successive coefficients. Doing so, crossover does not create new compatibility coefficients but simply exchanges coefficients between vectors. The task of producing new compatibility coefficients is therefore assigned to the mutation operator which consists of randomly reversing the value of every bit within a chromosome with fixed probability P_m.

Basically, GAs are maximization procedures as they tend to favor high-fitness individuals but we are dealing with a minimization problem. Therefore, a mapping is required between the objective function E defined in the previous section and the fitness function F. To do this, we used the following formula proposed by Caudell and Dolan (1989) in their experiments of neural network learning:

$$F = \tan\{\tfrac{1}{2}\pi(1 - E/E_{\max})\} \tag{7}$$

where $E_{\max}$ is the maximum value of E (i.e., $\Sigma_\gamma n_\gamma$). Notice that minimizing E is equivalent to maximizing F, because F is a decreasing function of E. This formula has the advantage of strongly favoring good individuals in the reproduction phase; in fact, F tends to infinity as E goes to zero and near-optimal individuals are therefore allocated a very high probability of being reproduced into the successive generation, according to (6).

One problem that may arise when using GAs is premature convergence toward mediocre individuals. This is generally caused by early domination of a few extraordinary members in a mediocre population. In addition, frequently during the course of the optimization process, the population's average fitness is close to the population's best fitness; this is an undesirable behavior for the GA because average and best individuals will have nearly the same number of copies in next generations. In order to avoid such situations, Goldberg (1989) suggests scaling the fitness function as $F' = aF + b$ where a and b are appropriate parameters determined so that (1) the average scaled fitness F'_{avg} equals the raw average fitness F_{avg} and, (2) $F'_{\max} = kF_{\text{avg}}$, where $F'_{\max}$ is the scaled maximum fitness value, and k is the desired expected number of copies for the best population member. Following Goldberg's suggestion, in our experiments a value $k = 2$ was chosen.

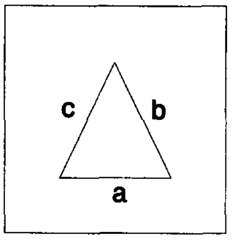
In contrast with the gradient algorithm, the GA-based learning procedure is computationally much less demanding as it requires, for each generation, a number of operations roughly proportional to Sm^2 (this should be compared with the $O(m^4)$ operations required by the gradient procedure). This is in fact the number of calculations needed to compute the fitness function, as seen in formulas (1)–(4), and to perform both the mutation and the crossover operations. However, due to the stochastic nature of the algorithm and the diversity of the operations performed during a genetic search (i.e., bit reversing, substring swapping, etc.), a precise estimation of the computational complexity of the GA-based learning algorithm is not as simple as for the gradient procedure. Moreover, GAs typically need many generations to find an optimal solution and this suggests that, as far as learning time is concerned, the GA learning procedure cannot be regarded as a serious competitor of the gradient method when $S \approx m^2$.

4. Results

In order to assess the effectiveness of the proposed evolutionary learning algorithm, some experiments on both a toy and a practical application were carried out. The first task involved labeling the sides of a triangle, whereas the second one consisted of tagging words with their parts-of-speech. In both applications we made a comparison between the genetic and the gradient learning algorithm. Specifically, the GA was run using the following parameters, which appeared to be nearly optimal: $S = 50$, $P_c = 0.5$, $P_m = 0.01$ for the triangle example and $P_m = 0.001$ in the part-of-speech task; furthermore, in both applications each compatibility coefficient was encoded into a 10-bit string. As to the gradient algorithm, the step size α_k in formula (5) was kept fixed at 0.1. Moreover, in the triangle example the direction vector $\boldsymbol{u}_k$ was normalized to avoid unacceptable oscillations of the algorithm, a behavior that was not observed in the part-of-speech task.

4.1. Labeling a triangle

This is the standard toy application for the relaxation labeling process, originally introduced by Rosenfeld et al. (1976) to practically study the dynamical behavior of the algorithm. The problem consists of labeling the sides of a triangle on a background, according to a 3-D interpretation. Each side can be interpreted either as a convex (+) or concave (−) dihedral angle with both faces visible, or as a dihedral angle with only one face visible, i.e., an occluding edge (< or >). We note that in the latter case there are two different labels because the visible face can be on either side of the edge: the direction of the arrow determines which side of the edge corresponds to the visible face. Overall, there are $4^3 = 64$ possible labeling configurations but only



	a	b	c
1.	>	>	>
2.	<	<	<
3.	>	-	>
4.	>	>	-
5.	-	>	>
6.	<	+	<
7.	<	<	+
8.	+	<	<

Fig. 1. A triangle and its meaningful interpretations.

eight of them correspond to meaningful interpretations, as shown in Fig. 1.

In the experiments presented here we idealized a very unfavorable as well as unlikely scenario for the learning algorithm in that the initial labeling assignments were strongly biased toward incorrect yet meaningful interpretations. Fig. 2 shows the learning set used in the study; the goal was to train the relaxation process to produce the desired labelings, given the input labelings, after a predetermined number of iterations. In particular, to study how the number of iterations performed by the relaxation labeling process affects the learning abilities of the competing algorithms, three series of simulations were carried out by fixing them at 1, 5, and 10. It is readily seen that standard statistical compatibilities (like correlation or mutual information) perform poorly on such a task for they tend to strongly favor the meaningful (yet incorrect) interpretations. This can be easily explained by observing that statistical compatibilities do not incorporate any information regarding the desired labeling assignments, being derived solely on the basis of an available set of consistent labeling configurations. Owing to such objective difficulties we found this problem to be an attractive benchmark for evaluating the performance of relaxation labeling learning procedures.

For each predetermined number of relaxation labeling iterations (i.e., 1, 5, and 10), ten independent runs of both the gradient and the genetic algorithm were performed, each started from randomly chosen initial compatibility vectors. The performance of the algorithms is shown in Fig. 3, where the mean learning curves along with their standard deviations (based on the ten trials) are plotted. More precisely, in the case of the GA the graphs represent the average behavior of the *best* population members in each generation. As can be clearly seen, in all the three series of experiments the GA performed significantly better than the gradient algorithm, and exhibited also a much more robust behavior. It is interesting to notice that this superiority becomes more and more marked as the number of relaxation labeling iterations increases. This behavior has an intuitive explanation. Observe, in fact, that allowing the relaxation labeling process to perform more cycles amounts to increasing the degree of nonlinearity in the error function E; consequently, the surface defined by E becomes much more complex and this enhances the tendency of the gradient algorithm to get stuck in local minima (cf. Fig. 3(c)).

Additionally, the computational cost of the two learning procedures was compared. Each gradient iteration was found to be from 1.5 to 2.6 times faster than the corresponding GA-generation. This is not surprising since, as seen in the previous section, the GA is expected to be slower than the gradient algorithm when the number of labels is small (here we

Case	Initial labeling				Desired labeling			
	>	<	-	+	>	<	-	+
	.30	.70	.00	.00	1	0	0	0
1)	.30	.70	.00	.00	1	0	0	0
	.30	.70	.00	.00	1	0	0	0
	.70	.30	.00	.00	0	1	0	0
2)	.70	.30	.00	.00	0	1	0	0
	.70	.30	.00	.00	0	1	0	0
	.30	.70	.00	.00	1	0	0	0
3)	.00	.00	.30	.70	0	0	1	0
	.30	.70	.00	.00	1	0	0	0
	.30	.70	.00	.00	1	0	0	0
4)	.30	.70	.00	.00	1	0	0	0
	.00	.00	.30	.70	0	0	1	0
	.00	.00	.30	.70	0	0	1	0
5)	.30	.70	.00	.00	1	0	0	0
	.30	.70	.00	.00	1	0	0	0
	.70	.30	.00	.00	0	1	0	0
6)	.00	.00	.70	.30	0	0	0	1
	.70	.30	.00	.00	0	1	0	0
	.70	.30	.00	.00	0	1	0	0
7)	.70	.30	.00	.00	0	1	0	0
	.00	.00	.70	.30	0	0	0	1
	.00	.00	.70	.30	0	0	0	1
8)	.70	.30	.00	.00	0	1	0	0
	.70	.30	.00	.00	0	1	0	0

Fig. 2. Training set used in the toy-triangle example.

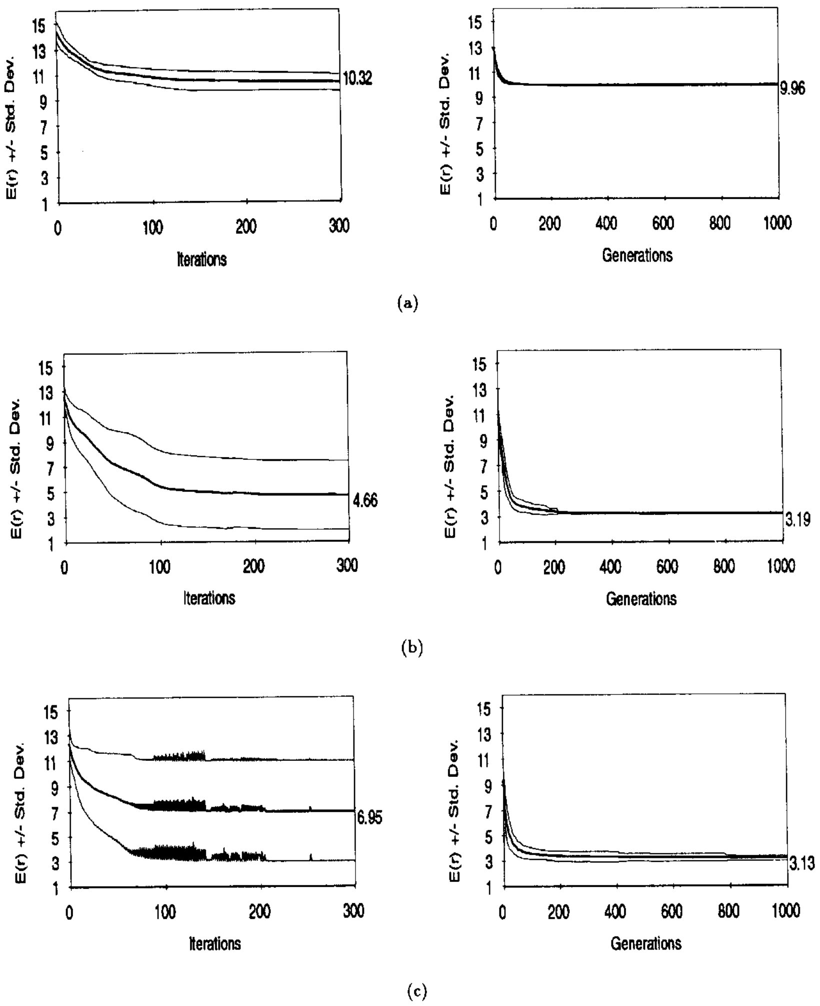


Fig. 3. Behavior of the mean error function E ($\pm$ standard deviation) during training for the triangle example. Left: gradient algorithm; Right: GA. The graphs from (a) to (c) correspond to 1, 5, and 10 relaxation labeling iterations, respectively.

have $m = 4$). It is a remarkable fact, however, that the GA was always capable of finding nearly optimal solutions within few generations.

4.2. Part-of-speech disambiguation

Labeling words according to their parts-of-speech is a fundamental problem that is encountered in many different contexts such as, for example, speech recognition, speech synthesis, and character/text recognition (Derouault and Mérialdo, 1984; Church, 1989; Plamondon et al., 1994). The problem is typically approached by two consecutive steps. In the first, each word within a sentence is associated with a list of potential labels; this can be accomplished by means of word-ending rules and/or a dictionary look-up. Due to the presence of homographs (i.e., words belonging to more than one syntactic class) a second step is needed wherein a disambiguation is carried out on the basis of context; this can be accomplished by a relaxation labeling process (Pelillo and Refice, 1991, 1994). Here, the objects to be labeled are words, the labels are the parts-of-speech, and the compatibility model expresses the degree of agreement between nearby syntactic classes.

In the experiments reported below, the label set contained the following main parts-of-speech: verb, noun, adjective, adverb, determiner, conjunction, preposition, pronoun, and a special miscellaneous label. Also, the context window employed for disambiguating a given word consisted of the position just after that word. Two separate 1,000-word sample texts were derived, one for training (26 sentences) and the other for testing (37 sentences). Both were extracted from a larger labeled corpus containing sentences taken from some issues of the EEC Italian Official Journal (Boves and Refice, 1987). The initial labeling assignments were obtained using a dictionary look-up which provided, for each word, the list of its possible labels. The dictionary, being constructed from the large corpus which included both the training and the testing set, had a 100% coverage. This is however a quite unrealistic simplication for in all non-trivial applications it is unlikely that any computerized dictionary, whatever its size, will contain all the words found in unrestricted texts. This is especially true for Italian which, unlike English, is a strongly inflected language. This observation motivated us to remove from our original complete dictionary a small percentage of the less frequent words, so that the resulting coverage of both training and testing words became approximately 96% (this was experimentally found to be the expected asymptotic coverage of Italian dictionaries (D'orta et al., 1988)). Accordingly, to derive the initial labeling assignments the following rule was adopted. Each word within a sentence was first searched for into the dictionary; if the search succeeded, then all its potential labels were given uniform probability. Otherwise, the probability mass was uniformly distributed among the following "open" syntactic classes: verb, noun, adverb, and adjective (in that case, in fact, the word could not be in any of the remaining "closed" classes, whose representatives were all contained into the dictionary). Doing so, the training and the testing texts were found to have 170 and 158 ambiguous words (i.e., having more than one candidate label), respectively.

Two series of experiments were carried out. In one, the relaxation labeling process was stopped after the first iteration, whereas in the other was allowed to perform exactly five iterations. In both, ten GA as well as gradient-based learning sessions were performed, each started with random initial configurations. The average behavior of the error function E and the corresponding disambiguation accuracy (i.e.,

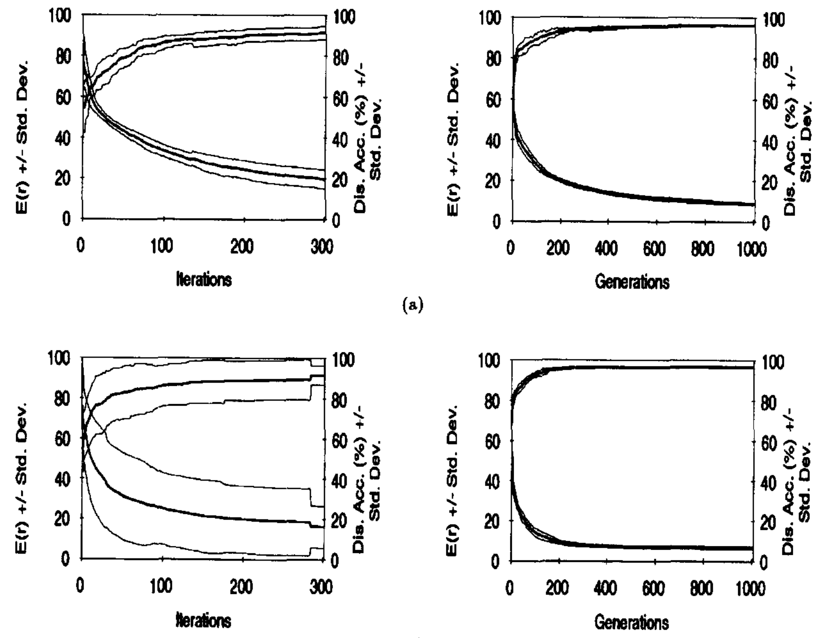


Fig. 4. Behavior of the mean error function E and disambiguation accuracy ($\pm$ standard deviations) during training for the part-of-speech example. Left: gradient algorithm; Right: GA. (a) 1 relaxation labeling iteration; (b) 5 relaxation labeling iterations.

Table 1
Mean disambiguation accuracy (± standard deviation) over the test set for the part-of-speech example, using the best compatibilities found by the gradient and the genetic procedures

Number of relaxation labeling iterations	Gradient	GA
1	80.19% (±1.37)	84.49% (±1.50)
5	79.49% (±2.96)	85.38% (±1.94)

the percentage of correctly disambiguated ambiguous words), along with their standard deviations is shown in Fig. 4. As for the triangle example, the GA did a better job than the gradient procedure of finding good compatibility coefficients, and exhibited also a more stable behavior. In this case, too, the difference in performance becomes more evident when the relaxation labeling algorithm is allowed to perform more iterations.

Next, to evaluate the generalization performance, the "best" compatibilities found by both algorithms in each of the training session were employed to run the relaxation labeling process over the 1,000-word test set. Table 1 summarizes the results obtained.

As can be seen, the GA obtained better results than its competitor in both series of experiments. Interestingly, among the best compatibility vectors found in the GA training trials, the highest generalization rate was obtained by the one having the poorest training-set performance; this seems to indicate that "overfitting" may have taken place, a phenomenon that is widely documented in the neural network literature (Chauvin, 1990; Weigend et al., 1990). As for neural networks, the use of "cross-validation" (Weigend et al., 1990) can thus help prevent relaxation labeling processes from being overtrained, thereby resulting in better generalization.

Finally, with regard to computational time, each GA-generation was found to be about four time faster than the corresponding gradient iteration. This demonstrates how the GA becomes a more practical approach than the gradient method when the number of labels increases. Like in the previous toy example, it is also to be noticed that the GA was able to achieve near-optimal results just after a few hundred generations.

5. Conclusions

We have proposed the use of genetic algorithms to learn the compatibility coefficients of relaxation labeling processes. Genetic algorithms exhibit several advantages over traditional gradient methods which make them particularly attractive for our learning problem; specifically, they are able to avoid local optima, do not require derivative information, and involve simple arithmetic operations. The experimental results have demonstrated that a simple GA performs substantially better than gradient-based learning procedures and turns out to be much more robust. It can be concluded therefore that GAs represent a reliable and effective technique for training relaxation labeling processes and, owing to their intrinsic simplicity, are particularly suited for real-world high-dimensional applications, where the gradient method would become impracticable.

References

Boves, L. and M. Refice (1987). The linguistic processor in a multi-lingual text-to-speech and speech-to-text conversion system. *Proc. Europ. Conf. Speech Technol.*, Edinburgh, Scotland, 385–388.

Caudell, T.P. and C.P. Dolan (1989). Parametric connectivity: Training of constrained networks using genetic algorithms. *Proc. 3rd Internat. Conf. Genetic Algorithms*, George Mason University, 370–374.

Chauvin, Y. (1990). Dynamic behavior of constrained back-propagation networks. In: D.S. Touretzky, Ed., *Advances in Neural Information Processing Systems 2*. Morgan Kaufmann, San Mateo, CA, 642–649.

Church, K.W. (1989). A stochastic parts program and noun phrase parser for unrestricted text. *Proc. IEEE Internat. Conf. Acoustics, Speech, Signal Processing*, Glasgow, Scotland, 695–698.

Davis, L. (Ed.) (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Davis, L.S. and A. Rosenfeld (1981). Cooperating processes for low-level vision: A survey. *Artificial Intell.* 17, 245–263.

Derouault, A.M. and B. Mérialdo (1984). Language modeling at the syntactic level. *Proc. 7th Internat. Conf. Pattern Recognition*, Montreal, Canada, 1373–1375.

D'orta, P., M. Ferretti, A. Martelli, S. Melecrinis, S. Scarci and G. Volpi (1988). Large-vocabulary speech recognition: A system for the Italian language. *IBM J. Res. Develop.* 32 (2), 217–226.

Eiben, A.E., E.H.L. Aarts and K.M. Van Hee (1991). Global convergence of genetic algorithms: A Markov chain analysis. In: H.-P. Schwefel and R. Männer, Eds., *Parallel Problem Solving from Nature*. Springer, Berlin, 4–12.

Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

Grefenstette, J.J. (1986). Optimization of control parameters for genetic algorithms. *IEEE Trans. Syst. Man Cybernet.* 16 (1), 122–128.

Holland, J.H. (1992). *Adaptation in Natural and Artificial Systems*, 2nd edition. MIT Press, Cambridge, MA.

Hummel, R.A. and S.W. Zucker (1983). On the foundations of relaxation labeling processes. *IEEE Trans. Pattern Anal. Mach. Intell.* 5 (3), 267–287.

Kullback, S. (1959). *Information Theory and Statistics*. Wiley, New York.

Lin, J. (1991). Divergence measures based on the Shannon entropy. *IEEE Trans. Inform. Theory* 37 (1), 145–151.

Peleg, S. and A. Rosenfeld (1978). Determining compatibility coefficients for curve enhancement relaxation processes. *IEEE Trans. Syst. Man Cybernet.* 8 (7), 548–555.

Pelillo, M. (1994). Nonlinear relaxation labeling as growth transformation. *Proc. 12th Internat. Conf. Pattern Recognition*, Jerusalem, Israel, 201–206.

Pelillo, M. and M. Refice (1991). Syntactic category disambiguation through relaxation processes. *Proc. Eurospeech '91*, Genova, Italy, 757–760.

Pelillo, M. and M. Refice (1994). Learning compatibility coefficients for relaxation labeling processes. *IEEE Trans. Pattern Anal. Mach. Intell.* 16 (9), 933–945.

Plamondon, R., S. Clergeau and C. Barrière (1994). Handwritten sentence recognition: From signal to syntax. *Proc. 12th Internat. Conf. Pattern Recognition*, Jerusalem, Israel, 117–122.

Rosenfeld, A., R.A. Hummel, and S.W. Zucker (1976). Scene labeling by relaxation operations. *IEEE Trans. Syst. Man Cybernet.* 6 (6), 420–433.

Rudolph, G. (1994). Convergence analysis of canonical genetic algorithms. *IEEE Trans. Neural Networks* 5 (1), 96–101.

Weigend, A.S., B.A. Huberman and D.E. Rumelhart (1990). Predicting the future: A connectionist approach. *Internat. J. Neural Syst.* 1 (3), 193–209.

Zucker, S.W., Y.G. Leclerc and J.L. Mohammed (1981). Continuous relaxation and local maxima selection: Conditions for equivalence. *IEEE Trans. Pattern Anal. Mach. Intell.* 3 (3), 117–127.