TEACHING RELAXATION LABELING PROCESSES USING GENETIC ALGORITHMS

Marcello Pelillo **Fabio Abbattista** **Angelo Maffione**

Dipartimento di Informatica, Università di Bari
Via E. Orabona 4, 70126 Bari, Italy

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. 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 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.

1. INTRODUCTION

Relaxation labeling processes are a broad class of popular techniques within the pattern recognition and machine vision domains [1]-[3]. 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. Over the past years, several heuristic statistical-based interpretations such as correlation [1] or mutual information [4] have been formulated.

Recently, a novel approach for determining the compatibilities of relaxation labeling procedures has been introduced which views the problem as one of *learning* [5]. This amounts to minimizing a certain cost function 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. In previous work [5], classical gradient techniques were used to accomplish this. However, gradient-based learning algorithms exhibit some inherent limitations that could prevent them from being applied to high-dimensional problems of practical interest. These include the inability to escape from local minima and their high computational complexity – which is of the order of the fourth power of the number of labels (or classes) of the problem at hand [5]. In addition, we note that some relaxation schemes are even non-differentiable (see, e.g., [6]) 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) [7]. We found GAs to be advantageous not only because are able to find nearly globally optimal solutions without being trapped into local optima, but also because are less computationally expensive, requiring on each step a time roughly proportional to the square of the number of labels.

2. RELAXATION LABELING AND THE LEARNING PROBLEM

Relaxation labeling processes involve a set of objects $B = \{b_1, \cdots, b_n\}$ and a set of labels $\Lambda = \{1, \cdots, m\}$. The purpose is to label each object of B with one label of Λ. By means of some local measurement it is generally possible to construct, for each object b_i, a vector $\mathbf{p}_i^{(0)} = (p_{i1}^{(0)}, \cdots, p_{im}^{(0)})^T$ such that $p_{i\lambda}^{(0)} \geq 0$ (all i and λ) and $\sum_\lambda p_{i\lambda}^{(0)} = 1$ (all i). Each $\mathbf{p}_i^{(0)}$ can therefore be interpreted as the *a priori* (non contextual) probability distribution of labels for b_i. By simply concatenating $\mathbf{p}_1^{(0)}, \mathbf{p}_2^{(0)}, \cdots, \mathbf{p}_n^{(0)}$, we obtain an initial weighted labeling assignment for the objects of B that will be denoted by $\mathbf{p}^{(0)} \in \mathbb{R}^{nm}$. The compatibility model is

D. W. Pearson et al., *Artificial Neural Nets and Genetic Algorithms*

represented by a four-dimensional matrix of real-valued nonnegative compatibility coefficients R: the element $r_{ij}(\lambda,\mu)$ measures the strength of compatibility between the hypotheses "λ is on object b_i" and "μ is on object b_j." In what follows, we will find it convenient to "linearize" the compatibility matrix and consider it as a column vector $\mathbf{r}$.

The relaxation labeling algorithm accepts as input the initial labeling assignment $\mathbf{p}^{(0)}$ and updates it iteratively taking into account the compatibility model, in order to achieve global consistency. At the tth step ($t = 0,1,2,\cdots$) the labeling is updated according to the following formula [1]:

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

where

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

The process is stopped when some termination condition is satisfied (e.g., when the distance between two successive labelings becomes negligible) and the final labeling is usually used to label the objects of B according to a maxima selection criterion [6].

Now, let us focus on the learning problem. Suppose that a set of learning samples $L = \{L_1,\cdots,L_N\}$ is given, where each sample L_γ is a set of labeled objects of the form

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

For each $\gamma = 1\ldots N$, let $\mathbf{p}^{(L\gamma)} \in \mathbf{R}^{n_\gamma m}$ 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 $\mathbf{p}^{(I\gamma)}$ on the basis of the objects in L_γ, and let $\mathbf{p}^{(F\gamma)}$ denote the labeling produced by the relaxation algorithm when $\mathbf{p}^{(I\gamma)}$ is given as input.

Broadly speaking, the learning problem for relaxation labeling is to determine a compatibility vector $\mathbf{r}$ so that the final labeling $\mathbf{p}^{(F\gamma)}$ be as close as possible to the desired labeling $\mathbf{p}^{(L\gamma)}$, for each γ. To do this, we can define a cost function measuring the loss incurred when $\mathbf{p}^{(F\gamma)}$ is obtained instead of $\mathbf{p}^{(L\gamma)}$, and attempt to minimize it. Here, we use the following information-theoretic divergence measure recently proposed by Lin [8]:

$$E_\gamma = n_\gamma - \sum_{i=1}^{n_\gamma} \log_2\left(1 + p_{i\lambda_i^\gamma}^{(F\gamma)}\right) . \tag{3}$$

The total error over L can therefore 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 $\mathbf{r}$. In [5], this problem is solved by means of a gradient method which begins with an initial point $\mathbf{r}_0$ and produces a sequence $\{\mathbf{r}_k\}$ as follows: $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k \mathbf{u}_k$, where $\mathbf{u}_k$ is a direction vector determined from the gradient of E, and α_k is a suitable step size.

3. LEARNING COMPATIBILITY COEFFICIENTS WITH GENETIC ALGORITHMS

Genetic algorithms are parallel search procedures largely inspired from the mechanisms of evolution in natural systems [7]. They work with a constant-size population of chromosomes or individuals, each associated with a fitness value that determines its probability of surviving at the next generation. In the present application, each chromosome represents a compatibility vector $\mathbf{r}$; each coefficient $r_{ij}(\lambda,\mu)$ is mapped into a fixed-length string of bits, and the whole chromosome is then obtained by simply 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 the reproduction operator, commonly referred to as *roulette-wheel* selection [7], consists of choosing the chromosomes that are to be copied in the next generation according to a probability proportional to its fitness. One problem with this mechanism is that the best individuals need not survive in future generations and this can slow down the convergence of the algorithm. To overcome this drawback, we made use of an *elitist* reproduction mechanism [9] 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. Once that the best individuals have been selected, the crossover operator is applied between pairs of individuals in order to