Towards on an optimized parallel KNN-FUZZY classification approach

J. L. A. Rosa, N. F. F. Ebecken & M. C. A. Costa
COPPE / Universidade Federal do Rio de Janeiro, Brazil

Abstract

This paper presents a classification method based on the KNN-Fuzzy classification algorithm, optimized by a Genetic Algorithm in a pc-cluster parallel environment. Analyses are made upon the results obtained in the classification of a large example in order to demonstrate the proposed approach. The performance assessment is also discussed.

1 Introduction

Data classification is one of the most used data mining tasks. Anderberg [1] defines classification as being the process or act to associate a new item or comment to a category. As example, a person can be classified, according some attributes: sex (female or male), nationality (country where it was born), naturalness (state where it was born), instruction degree (illiterate or not), height (low, high).

Most of the knowledge discovery techniques are based on mathematical statistics and Machine Learning fundamentals [2]. Many classification methods are used and some of them are described in [3].

A memory based reasoning strategy was adopted in this paper. Basically it needs: the set of stored cases, a distance metric to compute distances between cases and the number of nearest neighbors to retrieve. This corresponds to a classical non-parametric architecture.

The basic idea is very straight-forward. For training, all input-to-output pairs in the training set are stored into a database. When an estimation or a classification is needed on a new input pattern, the answer is based on the nearest training patterns in the database.
The method requires no training time other than the time required to preprocess and store the entire training set. It is very memory intensive since the entire training set is stored. Classification/estimation is slow since the distance between input pattern and all patterns in the training set must be computed.

This method typically performs better for lower dimensional problems (less than 10), as theoretical and empirical results suggest that the amount of training data needed in higher dimensions is greater than that required in other models.

One of the difficulties that arises when utilizing this technique is that each of the labeled samples is given equal importance in deciding the class memberships of the pattern to be classified and there is no indication of its "strength" of memberships in that class. This problem can be addressed by incorporating fuzzy set theory.

The basis of the fuzzy algorithm is to assign membership as a function of the vector's distance from its nearest neighbors and those neighbors' memberships in the possible classes.

The main questions now are:
- How many nearest neighbors (K) must be considered? (empirical formulas, K should be less than the square root of the total number of training patterns)
- How weight the distance to calculate each neighbor's contribution to the membership value?

In this paper those questions are answered by using a genetic algorithm. A parallel version was implemented, including all the commented improvements to build a more precise and efficient classification methodology.

Section 2 presents the KNN-Fuzzy Method and the genetic algorithm approach used to optimize it. The parallelization schema and its consequences, and the available computational environment are discussed in section 3. The studied case is analyzed with the Amdahl's law point of view in section 4. Some conclusions and comments are presented in section 5.

2 KNN-fuzzy method

KNN (K-Nearest Neighbors) consists of the identification of groups of individuals with similar features and its posterior grouping.

\[
\text{dist}_{i} = \sum_{r=1}^{K} \left( v_{u_ij} - v_{k_ij} \right)^2
\]

The calculation of the distance of the unknown sample in relation to the known samples is made by:

Assuming i known samples with two classes, 0 and 1, the KNN algorithm can be summarized as indicated in Figure 1.

Each distance is related to the corresponding class in a vector of distances. After the calculation of all the distances, the K smaller distances are selected; the unknown sample then is classified as being of the same present class.
As shown in [2], the value of K must be that one that satisfies the function \( \max \) that defines the maximum value of neighbors (samples) \( K_{\text{max}} \), who are near the unknown sample, in relation to all the samples of the considered set, \( K_{\text{tot}} \):

\[
K_{\text{max}} = \max (K_{\text{tot}})
\]  

In the classification task, it must be stressed that a variable can have more contribution in the definition of the class than others. A possible solution may be to consider the weight of the variables in accordance with its degree of importance for the classification process. According to [1], the balance of the variables transforms the original space in a representation space that can be wider or with reduced number of dimensions, which gives better classification accuracy.

In accordance with the work presented in [4], the basis of the KNN-Fuzzy classification algorithm is to associate the relevancy of a vector to its next \( K \)-neighbors and those members in possible classes.

Let \( W = \{ Z_1, Z_2, ..., Z_c \} \) be the set of \( c \) prototype representing the \( c \) classes. Let \( \mu_i(x) \) the membership (to be computed) associated to vector \( x \). As seen in (3)

\[
\mu_i(x) = \frac{1 / ||x - Z_i||^{2/(m-1)}}{\sum_{r=1}^{c} (1 / ||x - Z_r||^{2/(m-1)})}
\]
the associate memberships of \( x \), are influenced by inverse of the distances of the neighbors and its \( c \) membership’s classes. The inverse of the distance serves as weight, taking in consideration how close or not the vector to be classified is from the classified vector.

The variable \( m \) determines how heavily is the weight attributed at a distance of each neighbor to the value of membership. If the value of \( m \) is equal two, then the contribution of each point of the neighborhood is considered reciprocal to the long-distance of the point to be classified. As \( m \) increases, it diminishes the contribution of each neighbor; however, the reduction of \( m \) (\( m > 1 \)), implies more strong contribution. Assuming \( i \) known samples with two classes, 0 and 1 the algorithm can be summarized as shown in Figure 2.

BEGIN

READ \( i \) known samples (ks)
READ unknown sample (us)
Compute each distance from \( us \) to \( ksi \)
ASSIGN both distance and class to a distance vector
SORT distance vector by distance (column 1)
SELECT \( K \) vector distance samples
Compute \( \mu(x) \) using (3)

END

Figure 2: KNN-Fuzzy algorithm.

2.1. The proposed methodology

In this paper, a standard genetic algorithm scheme is employed to evolve the KNN-Fuzzy method in order to achieve the best classification precision.

The binary coding [5] was used to generate the chromosome (gene). The representation corresponds to \( 2n \) chromosomes, where \( n \) is the amount of genes that each chromosome possesses. The set of all the patterns that the chromosome can assume forms its space of search. In the present study the \( K \), \( m \) and weight variables are taken into account.

To evolve the population the genetic operators promote modifications that improve the average performance of the population to each generation, using information on the previous generations. The new population starts with the selection of the individuals of the ascending population with bigger probability to compose the following generation. In a population of \( n \) individuals, each one with fitness \( f \), a chromosome \( c \) has it probability \( p \) of selection of this individual given by:
There are many selection methods, as described in [6]. In this work the method of stochastic selection via tournament was used. Among these chromosomes, the one with bigger aptitude is selected for the following population. The process then is repeated until it fills the descending population. Once selected the individuals of the descending population, the operators of crossover and mutation are applied. The crossover operator makes swap on fragments between pairs of chromosomes, whose strings are cut in a random position producing, for each individual, two amounts of alleles that will have to be reallocated.

Mutation inverts one or more bits of an individual in a probability generally low, improves the diversity of the chromosomes in the population and destroys the information contained in the chromosome.

In this way the values of K, m and the weights of the variables were obtained to achieve the best classification precision.

3 Parallel implementation

The main issues under consideration in the study of parallel implementation of conceived sequential algorithms are the partitioning algorithm schema and the target machine.

3.1. Partitioning algorithm schema

For the partitioning algorithm schema the study considers the use of the data parallelism approach that keeps a copy of the entire labeled samples, with the internal variables and functions, in each processing node partitioning training data set among nodes. This approach ensures that all values needed during the training phase like K, m and variable weights, are locally available, reducing the number of messages passing among nodes and the algorithm synchronization. In fact, all nodes perform only one communication after each complete presentation of a training data subset. In distributed memory machines this approach seems to be very efficient leading to great gain of performance. This implementation of data parallelism uses a control node that performs genetic operations, realizes chromosome encoding and decoding, broadcasts the values of K, m and weights of the variables, and summarize the interested value (total of right classes). All nodes start with same internal parameters but with different subsets of training data. During training phase of KNN-Fuzzy method, each node passes its training data subset producing partial errors that must be combined with partial errors produced by other nodes, generating an overall error. This error is used to update the weights of the variables in each node, until the procedure reaches the minimal...
acceptable overall error. It can be pointed out that each node only broadcasts its partial error to the control node at this time. All other calculations involve local data and can be made without synchronization. This drastically reduces the communications between nodes.

The application uses few and simple MPI standard primitives and can be portable for several machines.

3.2. Target machine

The machine available in this work was an academic pc-cluster of the NACAD/COPPE/UFRJ High Performance Computing Laboratory. This machine contains 8 nodes (Intel Pentium III with 256 MB memory) each connected by a Fast Ethernet network). The Linux operating system and LAM MPI [7] programming environment supports the parallel execution of jobs for that environment. LAM features a full implementation of the MPI communication standard among parallel process. The access obtained for the execution of the application in this environment was shared, i.e., all programs share all nodes during execution. It could be emphasized here that time tests were made with unbalanced work on nodes.

4 The studied case

The data set has been extracted from a real world insurance database that contains relationships among clients, contracts and tariffs. The former objective of this application is modeling classification criteria for two predefined classes and performance assessment. Some characteristics and values have been protected by non-disclosure policies. Despite of the relation's complexity, the attributes and its values were grouped in one single table for this study. This table contains 80 attributes and 147478 registers for the predefined classes. The data set had been pre-processed, clearing some aspects not suitable for data mining methods. The resultant set contains 64 attributes and 130143 lines of registers. The application in each node reads the data files (known, training and test sets), according to the "fagknmp.cfg" configuration file, which can be altered depending on desired data set partitioning. The application of Amdahl's Law for performance evaluation shows a 99.6% of parallelization coefficient for the developed algorithm. This is explained by the large time expended by KNN-Fuzzy procedure in the overall application time (in a serial evaluation the total computation time is equal 105057 s and the KNN-Fuzzy procedure evaluation time is equal 104634 s. In table 1 are presented the elapsed time spent in a limited execution of the application with different number of nodes for each job submission, the ideal (IdSp) and obtained speedup (Sp) and efficiency (Ep).

Figure 3 presents the graphical representation of the ideal and obtained speedup, according to Amdahl's Law.

Figure 4 presents the obtained efficiency.
Table 1: Jobs submission report.

<table>
<thead>
<tr>
<th>nodes</th>
<th>time (s)</th>
<th>IdSp</th>
<th>Sp</th>
<th>Ep</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>57804</td>
<td>1,991980</td>
<td>1,817469</td>
<td>0,995990</td>
</tr>
<tr>
<td>3</td>
<td>38287</td>
<td>2,976035</td>
<td>2,743934</td>
<td>0,992012</td>
</tr>
<tr>
<td>4</td>
<td>30173</td>
<td>3,952260</td>
<td>3,481821</td>
<td>0,988065</td>
</tr>
<tr>
<td>5</td>
<td>23945</td>
<td>4,920749</td>
<td>4,387430</td>
<td>0,984150</td>
</tr>
<tr>
<td>6</td>
<td>19905</td>
<td>5,881592</td>
<td>5,277920</td>
<td>0,980265</td>
</tr>
<tr>
<td>7</td>
<td>17775</td>
<td>6,834881</td>
<td>5,910380</td>
<td>0,976412</td>
</tr>
<tr>
<td>8</td>
<td>15880</td>
<td>7,780703</td>
<td>6,615680</td>
<td>0,972588</td>
</tr>
</tbody>
</table>

Figure 3: Ideal and obtained speedup.

Figure 4: Efficiency results.

5 Conclusions

The volume of data existing in real world data warehouse with complex relations can only be handled by methods implemented in high performance computers with efficient parallel methodology. The case used in this work has these characteristics and could only be handled by parallel application. The use of a standard library of communication primitives allows portability of the application.
for several machines, bringing great vantages for the methodology. The KNN-Fuzzy method, optimized by a genetic algorithm, has the ability of produce a very precise classification model comparing with other paradigms used in classification problems. The obtained performance of the parallel implementation has shown very good results.

References