© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

Data classification using KNN-fuzzy method optimized by a genetic algorithm

J. L. A. Rosa & N. F. F. Ebecken

COPPE / Universidade Federal do Rio de Janeiro / Brazil

Abstract

This paper presents a classification method based on the KNN-Fuzzy classification algorithm, supported by a Genetic Algorithm. It discusses how to consider data classification according to the Fuzzy logic and its consequences in Data Mining predictions. Analyses are made upon the results obtained in the classification of several databases in order to demonstrate the proposed approach.

1 Introduction

Data classification is one of the most used data mining tasks. ANDERBERGS [1] show many examples of classification in different fields as religion, etnia, type of job, style of clothes, etc., and also 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).

Small amounts of attributes, allow a simple and direct classification by principal component analysis and graphical analysis. 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].

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

2 KNN

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

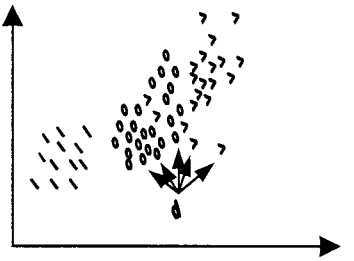


Figure 1- Comparison of unknown sample d for similarity with three preprocessed classes: \ , O e >.

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

$$\text{dist}_i = \sum_{j=1}^{j=m} (\text{vu}_{ij} - \text{vk}_{ij})^2 \qquad (1)$$

Assuming i known samples with two classes, 0 and 1, the KNN algorithm can be summarized as follows:

```
BEGIN
    READ i known samples (ks)
    READ unknown sample (us)
    Compute each distance from us to ks_i
    ASSIGN both distance and class to a distance vector
    SORT distance vector by distance (column 1)
    SELECT K vector distance samples
    Compare each class from the K selected samples
    IF (class is equal 0)
        Increment class0
    ELSE
        Increment class1
    ENDIF
END
```

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

Each distance is related to the corresponding class in a vector of distances. After the calculation of all the distances, as shown in Figure 2, the K smaller distances are selected; the unknown sample then is classified as being of the same present class.

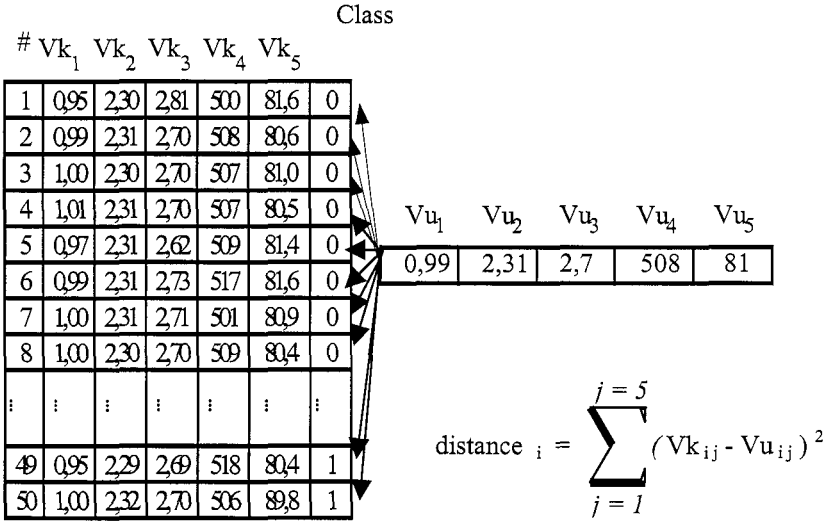


Figure 2 - Calculating de distance between unknown and known samples

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_{max}, who are near the unknown sample, in relation to all the samples of the considered set, K_{tot}:

$$K_{máx} = \max (K_{tot}) \quad (2)$$

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, that gives better classification accuracy.

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

3 Fuzzy sets

Adopting crisp sets concepts, in a subgroup of a set X, element x_i will only belong to the subgroup if it has 100% of the features of the elements of the subgroup, or either

$$x = \begin{cases} 0, & x \notin A \\ 1, & x \in A \end{cases} \tag{3}$$

A fuzzy set is a subgroup defined around an accurate set, which contains all the data of the considered universe. In the subgroup A including elements x of joint universe X, each element x is evaluated by the membership function: $\mu_A(x)$. This function represents any elements of X that belong to, total or partially, inside of the interval [0, 1]. If $\mu_A(x) = 1$, then x total belongs to the subgroup; if $\mu_A(x) = 0$, then the element does not belong to this subgroup. If $\mu_A(x) = 0.5$, then the element, has 50% of the features that define the elements of the subgroup. Thus, a fuzzy set is, defined around its elements and respective relevancies:

$$A = \mu_A(x_1) / x_1 + \mu_A(x_2) / (x_2) / x_2 + \ldots + \mu_A(x_n) / (x_n) \tag{4}$$

For example, given set X,

$$X = \{ x_1, x_2, x_3 \} \tag{5}$$

the subgroup of X defined as,

$$A = \{ 0.1 / x_1, 0.8 / x_2, 0.2 / x_3 \} \tag{6}$$

This expression says that x_1 has 10% of relevancy to the subgroup, x_2 has 80% of relevancy and x_3 has 20% of relevancy. It is observed that the addition of the relevancies does not result in 100%, because the relevancy of each element to the considered subgroup. In accordance with [4]: "As the complexity of a system magnifies, our ability to make necessary and significant comments on its behavior diminishes until a point that accuracy and significance (or relevance) become mutually exclusive characteristics". How much bigger the considered fuzziness (less accurate it will be the analysis of the data) bigger the amount of data considered classes. This explains low classification performance of some classification methods that does not consider predominant and not predominant characteristics.

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

4 KNN-Fuzzy

In accordance with the work presented in [5], 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 μ (x) the membership (to be computed) associated to vector x. As seen in (7), 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.

$$\mu_i(x) = \frac{1 / \| x - Z_i \|^{2/(m-1)}}{(1 / \| x - Z_j \|^{2/(m-1)})} \qquad (7)$$

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* greater than one), implies more strong contribution. Assuming i known samples with two classes, 0 and 1 the algorithm can be summarized as follow:

BEGIN
  READ *i* known samples (ks)
  READ unknown sample (us)
  Compute each distance from *us* to ks_i
  ASSIGN both distance and class to a *distance vector*
  SORT distance *vector* by *distance* (column 1)
  SELECT *K* vector distance samples
  Compute μ(x) using (7)
END

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

5 Genetic algorithm

In essence a GA is, according to GOLDBERG [7], an algorithm of fetching based on the mechanisms of the natural selection and on the genetics. GAs are inspired in the survival principle of the best individual, or either, better adequateness (fitness), which has greater probability to survive and to produce descendant for the following generation. To evaluate the produced solutions an objective function, called fitness, is used. In mathematical terms, the optimization consists of finding the solution that corresponds to the maximum of the objective function.

The binary coding [8] can be used to generate the chromosome (gene). The representation corresponds to 2^n 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. If the chromosome represents k parameters, then the space has k dimensions.

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. DE JONG and SPEARS [11] [12] [13] [14] [15] has studied the importance of each one of the genetic operators generally used in the evolution of a population. The formation of 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:

$$p_i = \sum_{i=1}^{N} f_i \qquad (8)$$

There are many selection methods, as described in [16]. In this work the method of the Selection for Match will be used, which randomly selects (with equal probabilities) n chromosomes of the population.

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 alelos 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.

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

In this work genetic algorithms are employed to find the values of K, m and the weights of the variables to achieve the best classification precision.

6 Results

This paper shows some results that corroborate the hypothesis that a KNN-fuzzy sorting can minimize the time of use of the CPU, finding, with the aid of a genetic algorithm, the necessary parameters for good/excellent sorting ratio.

Initially some usual benchmarks were processed.

The datasets and attributes (variables) used here are indicated in Table 1.

Table 1 - Data sets descriptions

Database	Samples	Variables	classes
Iris Plant Database [17]	150	4	3
Pima Indians Diabetes [17]	768	8	2
Balance Scale [17]	625	4	3
Liver-disorders [17]	345	6	2
Erudit [18]	770	27	2

The best classification results (%) and computation times are summarized in Table 2.

Table 2 - Classification results and computer times

Database	% precision	Time (s)
Iris Plant Database	100.0	3
Pima Indians Diabetes	89.0	80
Balance Scale	90.9	149
Liver-disorders	91.4	67
Erudit	94.4	393

To explicit the gain obtained by the GA optimization some results are indicated in table 3.

Table 3 – Comparison between standard and GA strategies

	Standard			GA		
	k	m	% precision	k	m	% precision
Iris	90	2	97	30	0.025781	100
Pima	384	3	79	56	0.0202148	89
Erudit	130	2	66	114	0.004297	94.4

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

The classification was performed in an IBM RS/6000 SP with 3 nodes, MIMD (Multiple Instruction Multiple Data) architecture connected via crossbar switch allowing direct connection between two nodes.

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

Two strategies have been used. In the first strategy, the database classified was divided between nodes, and the training and test bases had been copied for each node. After the calculation of the distances a parallel sort was done, causing an overhead in the classification process and, sometimes, hindering the convergence. In the second strategy, the classified basis was copied for each one of nodes, dividing per nodes the training and test bases. In this case, the process converged.

To prevent unnecessary overheads, the genetic operations and chromosome decryptation, was realized in node 0 (zero).

The base Insur was presented to the academic world in 1998, when it had a competition proposal in a congress [19]. This case uses data extracted from a private company of life insurance Data Warehouse. The original files given by the company describe the relationship between customers, contracts of insurance and components of tariff. The tables had been unified, generating a base with 80 columns and 147478 records. These records describe the customers, and the columns characterize its, including personal information and data on the policy. On many attributes, less or none information are divulgated. After the data preparation, the database size becomes 64 colunms and 130143 strings. The class label has only two labels.

In several experiments, the speed-up was near of 1.8.

The precison obtained with the KNN standard algorithm was about 76%. The improvement result by the optimized approach added 15% to this ratio.

7 Conclusion

For classification tasks, the KNN-fuzzy method showed to be very useful. It improves the accuracy and results in high classification precision, not obtained with crisp implementations.

The genetic algorithm optimization has great importance in the improvement of the accuracy, finding optimal values of K and m.

The generation of the population of chromosomes in a random form showed, in practical, to be more attractive than the exploitation of previous populations.

Depending on the generation of the population, values close or not the global minimum can occur, resulting in a variation in the processing efforts.

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

References

[1] ANDERBERG, MICHAEL R. *Cluster Analysis for Applications*. Academic Press, 1973.
[2] MITCHIE, D., SPIEGELHALTER, D. J., TAYLOR, C. C. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, 1994.
[3] MANLY, B. F. J. *Multivariate Statistical Methods. A Primer* , Chapman & Hall, London, 1986.
[4] ZADEH, L. Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst., Man, Cybern, vol. SMC-3, 1973.
[5] KELLER, J. M. , GRAY, M. R. , GIVENS JR. , J. A. , *A Fuzzy K-Nearest Neighbor* Algorithm, IEEE Transactions on Systems, Man, and Cybernetics, v. SMC-15, n. 4, pp. 258-260, July/August 1985.
[6] HOLLAND, J.H. *Adaptation in Natural and Artificial Systems*. MIT Press, 1975.
[7] GOLDBERG, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, 1989.
[8] HEATH, F.G. , *Origins of Binary Code*, Scientific American, v.227, n.2, pp.76-83, Agosto 1972.
[9] ROBERTSON, G.G., *Population Size in Classifier Systems*. In: Proceedings of the 5th International Conference on Machine Learning, pp. 142-152, Ann-Arbor, june 1988.
[10] KALLEL, L., SCHOENAUER, M., *Alternative Random Initialization in Genetic Algorithms*. In: Proceedings of the 7th International Conference on Genetic Algorithms, East Lansing (MI), july 1997.
[11] DE JONG, K.A., SPEARS, W.M., *An Analysis of the Interacting Roles of Population Size and Crossover in Genetic Algorithms*, Proceedings of the International Conference on Parallel Problem Solving from Nature, pp. 38-47, Dortmund (GER), october 1990.
[12] SPEARS, W.M., DE JONG, K.A., *On the virtues of Parameterized Uniform Crossover*, Proceedings of the 4th International Conference on Genetic Algorithms, pp. 230-236, La Jolla (CA), july 1991.
[13] SPEARS, W.M., ANAND, V.A., *A Study of Crossover Oiperators in Genetic Programming*, Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems, pp. 409-418, Charlotte (NC), october, 1991.
[14] DE JONG, K.A., SPEARS, W.M., *A formal Analysis of the Role of Multi-Point Crossover in Genetic Algortihms*, Annals of Mathematics and Artificial Intelligence Journal, v.5, n.1, pp. 1-26, 1992.
[15] SPEARS, W.M., *Crossover or Mutation?*, Proceedings of the 2th Foundations of Genetic Algorithms Workshop, pp. 221-237, Vail (CO), july 1992.
[16] ESPÍNDOLA, ROGÉRIO P. *Optimizing Classification Fuzzy Rules by Genetic Algorithms*, M.Sc.,COPPE/PEC, maio, 1999.
[17] Databases from UCI Machine Learning Repository obtained in http://www.ics.uci.edu/~mlearn/MLRepository.html

© 2002 WIT Press, Ashurst Lodge, Southampton, SO40 7AA, UK. All rights reserved.
Web: www.witpress.com Email witpress@witpress.com
Paper from: *Applications of High Performance Computing in Engineering VII*, CA Brebbia, P Melli & A Zanasi (Editors).
ISBN 1-85312-924-0

[18] Databases from European Network in Uncertainty Techniques Developments for Use in Information Technology - ERUDIT, obtained in http://www.erudit.de/erudit/index.htm

[19] KDD-Sisyphus. KDD Sisyphus Workshop – Data Preparation, Preprocessing and Reasoning for Real-World Data Mining Applications. Internet: http://research.swisslife.ch/kdd-sisyphus, 1998.