STUDIA UNIV. BABEŞ-BOLYAI, MATHEMATICA, XXXVI, 3, 1991

SOFTWARE FOR CLASSIFICATION

CRISTIAN LENART*

Received: August 4, 1991
AMS subject classification: 68T10

Rezumat. Software pentru clasificare. Articolul prezintă un sistem de programe destinat clasificării automate a unei colecţii de obiecte caracterizate prin valorile mai multor parametri. Programele au fost elaborate de autor şi se bazează pe o serie de algoritmi din literatură, precum şi pe unii originali. Principalele componente ale sistemului sînt: extractorul de caracteristici, clasificatorul ierarhic diviziv, clasificatorul neierarhic, clasificatorul bazat pe arborescenţa de acoperire minimală şi componenta destinată interpretării calitative a partiţiilor obţinute. Pentru fiecare componentă se prezintă structura şi funcţiile ei, algoritmii implementaţi, datele de intrare şi ieşire.

0. Introduction. The aim of this paper is to describe a program system designed for pattern preprocessing, classification and interpretation of data sampled from a non-homogeneous population. The programs, which belong to the author of the paper, implement classical algorithms as well as some original ones. The main components of the system are: the pattern preprocessor, the divisive hierarchical classifier, the single-level classifier, the minimum forest classifier and the component which enables a qualitative interpretation of the obtained partitions. The input of the system is the collection of objects to be classified, characterized by the values of d variables recorded within a usual text file.

1. The pattern preprocessor. This component performs the transformation of data from the original pattern space to the feature space. The output is also a text file in which s features

* *"BABEŞ-BOLYAI" University, Faculty of Mathematics, 3400 Cluj-Napoca, Romania*

($s \leq d$) for each object are recorded.

This program has several processing options: normalization of the original patterns, Mahalanobis distance, principal component analysis and combinations of the above options. The first processing type is a simple scaling of the original variables by the overall mean and standard deviation such that they become comparable. The second option implements the Mahalanobis distance by an appropriate coordinate transformation. The principal component analysis projects the original patterns onto the eigenvectors of the covariance matrix of parameters corresponding to the first s eigenvalues in their decreasing order. A given threshold indicates what percentage of the original information should be preserved after data compression.

2. **The divisive hierarchical classifier.** This component implements the fuzzy divisive hierarchical algorithm [2], and its corresponding hard version. These algorithms perform a hierarchical descendant classification, iteratively splitting the current fuzzy (hard) cluster into two fuzzy (hard) subclusters until the clustering degree of this binary partition [3] becomes less than a given threshold; in this case, the cluster is considered homogeneous.

The input of this classifier is the file containing the features of the objects to be classified. There are three output files: one containing the hard partition (in the fuzzy case, it is obtained by defuzzification), another containing the fuzzy partition (in the fuzzy case) and the last one containing the

prototypes of the partition clusters.

The structure of the classifier is presented in fig. 1. Module HIER performs the hierarchical classification. Module SPLIT implements the generalized Fuzzy c-Means algorithm with two clusters, which is used to split the current cluster. It consists of the reiteration of modules: CENTRE (computation of

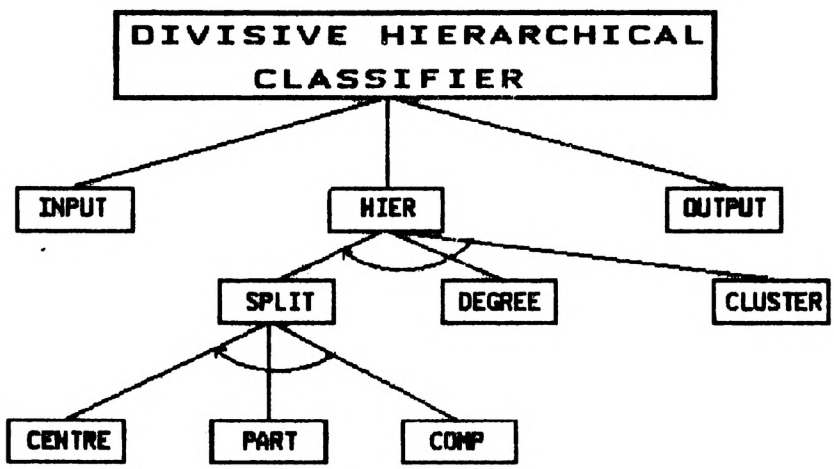

Fig.1

the sub-clusters centres), PART (computation of a new partition - fuzzy or hard) and COMP (comparison of the last two partitions). Module DEGREE computes the clustering degree of the current cluster binary partition. Module CLUSTER displays the clusters from the hierarchy and records within an output file final clusters (which are no longer splitted).

3. **The single-level classifier.** This classifier implements the following algorithms: Fuzzy c-Means, Fuzzy c-Lines, fuzzy clustering algorithms with linear manifold prototypes,

combinations of them [1] and hyperellipsoid prototypes [8], as well as their hard versions. This classifier performs a non-hierarchical classification, hence the number of clusters must be given by the user.

The input files of the classifier are: the file containing the features of the objects to be classified and a file containing an initial partition (fuzzy or hard) or an initial set of prototypes. Thus, the output of the divisive hierarchical classifier may become the input of this classifier, in order to obtain an improved partition. Moreover, if the input is a set of prototypes, this classifier may be used as a trainable classifier: the prototypes are computed from a training set and then unknown samples are classified according to the dissimilarities with respect to these prototypes. The output files are the same as those of the divisive hierarchical classifier.

The structure of this classifier is presented in Fig.2.

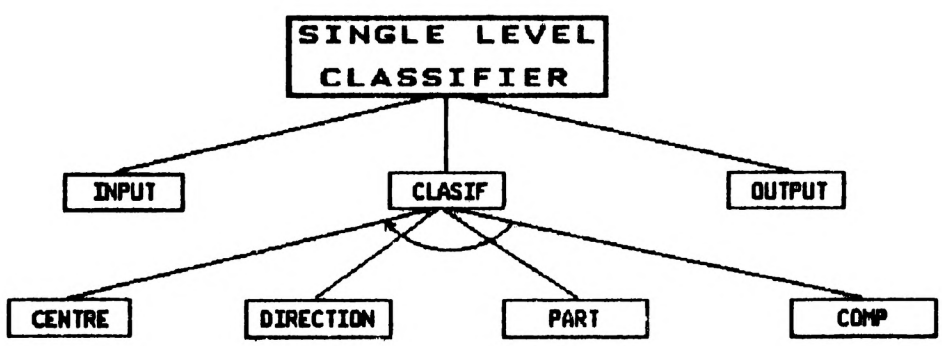

Fig.2

Module CLASIF performs the single-level classification. It consists of the reiteration of modules: CENTRE (computation of

the centres of the clusters), DIRECTION (computation of the directions of prototypes), PART (computation of a new partition - fuzzy or hard) and COMP (comparison of the last two partitions) until the last two partitions coincide (in the hard case) or the maximum difference of corresponding membership degrees does not exceed a given error level (in the fuzzy case).

4. **The minimum spanning forest (MSF) classifier.** From numerical experiments we noticed that Fuzzy c-Means and other related algorithms often misclassify samples situated at the border of the clusters. One way to prevent this situation, if the distribution of the cluster samples is close to a normal one, is to use hyperellipsoid prototypes. If the distribution is arbitrary, we propose Prim's MSF clustering algorithm [9] which is based on a graph-theoretical approach.

The MSF classifier first detects the subclusters containing samples which were surely correctly classified by a Fuzzy c-Means type algorithm and states them as "centres"; this operation, implemented in module SELECT (fig. 3), is done by selecting those samples which have the membership degree in the corresponding fuzzy cluster higher than a given threshold. The remaining samples represent the "objects" (according to the terminology used in [9]) and will be reclassified. Module DISSIM computes object-to-centre dissimilarities as point-to-set dissimilarities, i.e. the least dissimilarities between the objects and the samples in the centres. Module MSF, implementing Prim's MSF clustering algorithm, associates the objects one by one with the

centres. Thus, misclassified samples are reclassified and will probably get into the correct cluster.

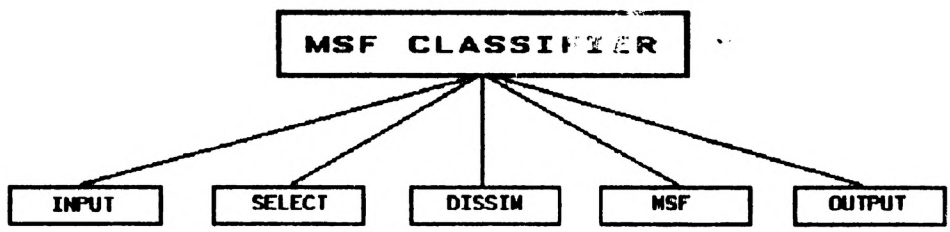

Fig.3

As input data we need the features of the samples, the fuzzy and the defuzzified partitions. The output is a hard partition.

5. **Interpretation of the obtained partitions.** We first give a theoretical model of partition qualitative interpretation. Let X be the set of samples partitioned into the clusters $A_1,\ldots,A_c$. Consider a qualitative feature (or a combination of qualitative features) F defined on X and taking a finite number of values $\{f_1, \ldots,f_k\}$ such that $k \geq c$. We are lookink for the onto function

$$\varphi : \{1,\ldots,k\} \to \{1,\ldots,c\}$$

which maximizes the cardinality of overlapping clusters from the initial partition of X and the one induced by the feature F:

$$S_\varphi = \sum_{i=1}^{k} | F^{-1}(f_i) \cap A_{\varphi(i)} |$$

This problem can be formulated as a maximum matching problem for which we apply the Hungarian algorithm. Thus the cluster A_j may

be interpreted as a mixture of the qualitative values from the set

$$\{f_i | \varphi\ (i) = j\}$$

We also define the matching degree as

$$\frac{S_\varphi}{|X|}$$

which is a sub-unitary value and characterizes the proportion in which the qualitative feature can explain the partition of X.

The component of our program system which enables the qualitative interpretation of obtained partitions enters quantitative and qualitative features of the classified samples fulfilling certain criteria concerning their features. Then a second module classifies the selected samples into groups as it was done by previous clustering procedures or according to the values of grouping features. Groups are identified as codes (for qualitative features) or intervals (for quantitative features). Quantitative features transformations can be performed. If two partitions are thus obtained, they can be compared as it was shown above. Selections, groups and transformed features can be stored in output files. Scatter plots of one quantitative feature against another can also be obtained and are useful in examining the performances of the clustering algorithms we used, the discrimination power of the two features and the regions delimited in the plane by the clusters.

6. **Facilities of the software.** This software gathers in a

unitary conception various aspects of classification: hierarchical and single-level classification, fuzzy and hard partitioning, pattern preprocessing and postprocessing, graph-theoretical methods and methods based on the minimization of a certain functional, supervised and unsupervised classification. Here are now some of the implementation facilities of this system:

- portability, as being written in Pascal language;
- simple text file structure of input and output data, which enables its use in a sequence of processing stages;
- independence of the system components, which permits interchanging their order during processing, omitting or reiterating them in order to obtain improved partitions;
- possibility of modifying the memory limits for data according to the capacity of the computer (this is done simply by modifying some constants and recompiling the programs);
- listing file option for obtaining a list with intermediate or final results;
- supplementary possibilities to limit the execution of iterative procedures by setting time, number of steps and maximum level in the classification hierarchy limits;
- other options which enable a flexible execution of programs.

This software was applied in geology, to the determination of certain types of mineralizations and to the parallelization of tuff horizons [5], in geography, to the regionalization of hydroenergetical potentials [6] and water resources [7] and in

biology, to the determination of plants associations specific to certain environment conditions [4].

REFERENCES

1. Bezdek,J.C., Coray,C., Gunderson,R., Watson,J., *Detection and characterization of cluster substructure*, SIAM J. of Appl. Math. 40 (1981) 339-371.
2. Dumitrescu,D., *Hierarchical pattern classification*, J. Fuzzy Sets and Systems 28 (1988) 145-162.
3. Dumitrescu,D., Lenart,C., *Hierarchical classification for linear clusters*, Studia Univ. "Babeş-Bolyai", Math. 3 (1988) 48-51.
4. Gardo,G., *Quantitative and qualitative structure of ligneous vegetation from the Făget forest Cluj-Napoca*, Thesis, "Babeş- Bolyai" Univ. Cluj-Napoca, 1991.
5. Ghergari,L., Lenart,C., Mârza,I., Pop,D., *Anorthitic composition of plagioclases, criterion for parallelizing tuff horizons in the Transylvanian basin*, to appear in Transylvanian Miocene Symposium.
6. Haidu,I., Lazăr,I., Lenart,C., Imbroane,A., *Modelling of natural hydroenergy organization of the small basins*, Proceedings of World Renewable Energy Congress, Sept. 1990, Reading, England, 3159-3167.
7. Haidu,I., Lenart,C., *Clustering techniques in water resources regionalization*, to appear in Annales Geophysicae.
8. Lenart,C., *A classification algorithm for ellipsoid form clusters*, "Babeş-Bolyai" Univ. Cluj-Napoca, Fac. Math. Res. Seminars 9 (1989) 93-102.
9. Lenart,C., *Graph-teoretical classification methods from a metrical point of view*, to appear in Discrete Mathematics, USA.
10. Tou,J.T., Gonzales,R.C., *Pattern recognition principles*, Addison-Wesley, 1981.