A Web-based Learning System for Statistical Pattern Recognition

J. LOBATO, J.S. SÁNCHEZ, and J.M. SOTOCA

Dept. Llenguatges I Sistemes Informàtics, Universitat Jaume I
Av. Sos Baynat s/n, 12071 Castelló (SPAIN) Tel.(+34) 964 728 350   Fax.(+34) 964 728 435
E-mail: jlobato@vision.uji.es, sanchez@uji.es, sotoca@uji.es

Abstract *– This paper presents a web-based learning system in support of a Ph.D. course in Statistical Pattern Recognition. More specifically, the learning system allows the user to know the main concepts related with several distance-based classification and prototype selection algorithms, as well as to interactively experiment with them by means of Java applets. The aim of such a web-based learning system is to show the behavior of several methods and techniques from the Statistical Pattern Recognition domain. From our experience, it has been possible to improve students learning by increasing interactivity.*

1. INTRODUCTION

Pattern Recognition is the research area that studies the operation and design of systems that recognize patterns in data. It encloses very distinct disciplines, such as discriminant analysis, feature extraction, error estimation, cluster analysis, grammatical inference and parsing (sometimes called syntactical pattern recognition). Important application areas are image analysis, character recognition, speech analysis, man and machine diagnostics, biometric identification, and industrial inspection. No single theory of Pattern Recognition can possibly cope with such a broad range of problems. However, there are several standard models, including: Statistical Pattern Recognition [1], Syntactic or Structural Pattern Recognition [2], Knowledge-based Pattern Recognition [3], and Adaptive Pattern Recognition and Neural Networks [4].

Many Ph.D. programs contain courses for each different Pattern Recognition approach. In particular, the program at Universitat Jaume I of Castelló includes three courses specifically addressed to Pattern Recognition: Statistical Pattern Recognition, Syntactic Pattern Recognition, and Neural Networks. This paper is related to the statistical approach and in this sense, pattern recognition is viewed as classification, that is, assigning an input or sample to a category or class.

The concepts studied in the Statistical Pattern Recognition course include topics with which students are not familiarized. Moreover, it can be difficult to understand the utility of several techniques and methods from this research domain. With the aim of showing how these methods work, web-based interactive learning systems can become a very useful tool for studying Statistical Pattern Recognition. In general, such systems provide a brief description of a particular subject and allow to perform interactive experiments in order to test different algorithms.

Within this context, we present in this paper a web-based learning system that is being employed in a Ph.D. course in Statistical Pattern Recognition at Universitat Jaume I. Such a learning system constitutes a complementary tool for studying several distance-based classification and prototype selection techniques.

From now on, the organization of this paper is as follows. Section 2 depicts some characteristics and advantages of web-based learning systems. Section 3 presents the distance-based classifiers that have been implemented in the web-based learning system. Section 4 reviews the prototype selection methods employed in such a system. Section 5 describes the learning system for the Ph.D. course in Statistical Pattern Recognition. Section 6 provides the main conclusions and draws some further extensions to this learning system.

2. WEB-BASED LEARNING ENVIRONMENTS

Web-based systems [5] have features that constitute a new learning environment for all involved: students, instructors, administrators, and parents. The system is designed for use both in the classroom as well as at home. For the student, it can provide continuous access to instructional materials and to learning and assessment tools. For the teacher, it can provide automated assessment and real-time reports that yield information needed to refine and improve instruction.

The system can individualize the learning environment at virtually any instructional level, by institution, by class or by student. It provides an immediate record of the learning status and progress of the students. Information needed to optimize the pace of instruction, to iterate topics and modify materials is readily available. At the institution level, it can provide greater instructional coherence among instructors. Beyond a specific school, it can provide greater coherence among institutions. The system has great potential to improve consistency in the instructional process and produce greater efficacy and efficiency in the learning process.

A web-based learning system is organized on client-server principles. In general, the server side of the system contains a database of learning materials, along with other components such as elements of active interaction or a database of assessment units.

From the client side, both the instructor and the student enter the system using standard browsers anytime and anyplace they can connect to the Internet. Thereby students extend their learning opportunities by accessing at any time exactly the same learning materials that they are using in the classroom.

The rapid development of Internet and World Wide Web now offer the possibility of conducting computer-based learning on a global scale, without the usual restrictions of platform dependence, the cost of mailing materials, and identifying interested users. Instead of searching for people who would like to use a given course, the Internet allows the student to find the material and review it at his own pace. Because web pages can be updated at any point, it is possible to keep the learning materials up to date without having to do anything except edit the *HTML* documents involved. It is also possible to keep channels of communication open between instructor and student. Finally, the use of server-side scripts and interactive forms make it possible for the student to be tested on what he has learned without any need for supervision.

This combination of factors makes the exploitation of web-based courses one of the most exciting uses for the Internet and World Wide Web, and a powerful tool for learning and intellectual advancement.

3. DISTANCE-BASED CLASSIFIERS

Classification has traditionally been tackled through two alternative approaches; namely, parametric and non-parametric methods [6]. The parametric classifiers assume a functional distribution of given samples. On the other hand, the non-parametric do not assume any functional distribution of the set of prototypes. While the parametric approach has theoretically been shown to be potentially capable of yielding optimal results, in practice, it often tends to actually fail because of inappropriate assumptions of a priori distributions.

Among non-parametric methods, those which are based on sample-to-sample distances are particularly remarkable; namely, *k*-NN techniques [7]. When applied to classification, these schemes require the classes to be represented by appropriate sets of prototypes and the decision rule is generally reduced to label each given sample with the class that contains most of its k nearest neighbors. It is the conceptual simplicity of such a rule, along with its asymptotical tendency towards the Bayes rule in terms of minimum classification error, what makes the *k*-NN approach particularly appealing in many practical situations. Nevertheless, when the number of prototypes in the training set is not large enough, the *k*-NN rule is no longer optimal. This problem becomes more relevant when having few prototypes compared to intrinsic dimensionality of the feature space, which is a very common practical situation.

Some alternative definitions of neighborhood have been used to obtain other non-parametric classifiers, trying to partially overcome the practical drawback pointed out above for the *k*-NN rule. In particular, the recently introduced concept of Nearest Centroid Neighborhood [8] along with the neighborhood relation derived from the Gabriel Graph (GG) and the Relative Neighborhood Graph (RNG) [9], have been used to obtain the so-called *k*-NCN and Graph Neighbors (GN) rules [10], respectively.

As mentioned before, the *k*-NN rule consists of estimating the class of a given sample through its *k* closest prototypes. In other words, this classifier considers that all the information required to classify a new sample can be obtained from a small subset of prototypes close to the sample. However, it does not take into account the geometrical distribution of those *k* prototypes with respect to the given sample (that is, in general the nearest prototypes do not completely surround the sample since the *k*-NN rule considers the neighborhood only in terms of a norm-based distance).

The non-parametric *k*-NCN and GN approaches are also based on the general idea of estimating the class of a sample from its neighbors, but considering a different kind of neighborhood which allows to inspect a sufficiently small area around the sample, in such a way that all prototypes surrounding that sample take part in the classification. As already pointed out, this is accomplished by using two different concepts about surrounding a sample with nearby prototypes: firstly, the Nearest Centroid Neighborhood [8], which tries to surround a sample by taking prototypes in such a way that (a) they are as near as possible to the sample, and (b) their centroid is also as close as possible to the sample.

Secondly, the Graph Neighborhood (i.e., GG and RNG-based neighborhoods) of a sample, defined as the union of all its graph neighbors. Taking into account that two points are graph neighbors if no other point lies inside a certain region of influence between them [9], it is possible to surround completely a sample by means of its graph neighbors.

Bearing this in mind, the Nearest Centroid Neighborhood and Graph Neighborhood concepts can be used to obtain two alternative non-parametric classifiers; namely, the *k*-NCN and GN rules, respectively. Both of them have in common the fact of considering a number of prototypes around (instead of close to) a sample to estimate its class. Given a set of labeled prototypes $X = \{x_1, ..., x_n\}$ and a new unknown sample *q*, the *k*-NCN and GN classification rules assign to *q* the class with majority of votes among its *k* nearest centroid neighbors or its graph neighbors, respectively.

4. PROTOTYPE SELECTION

The distance-based classifiers also present some drawbacks. First, the number of prototypes available is usually not enough to achieve the expected asymptotic performance. Second, the set of prototypes may contain noisy or erroneously labeled prototypes which usually lead to a decrease in performance. The resulting classifier needs, in principle, to store and search through the whole set of prototypes, operations that may imply some practical problems.

Prototype selection techniques have been proposed as a way of minimizing these drawbacks. They consist of selecting a particular subset of prototypes and applying the 1-NN rule using only the selected prototypes. Two different families of prototype selection methods exist in the literature. First, condensing algorithms aim at selecting the minimal subset of prototypes that leads to (approximately) the same performance than the 1-NN rule using the whole set. And second, editing algorithms eliminate erroneously labeled prototypes from the original set and “clean” the possible overlapping among different classes. This usually leads to significant improvements in performance.

4.1 Editing algorithms

The heuristic nature of most condensing algorithms contrasts with the strong statistical foundation of the most popular edited NN rules. In fact, the well-known Multiedit algorithm [11] is asymptotically optimal in the sense of Bayes. However, editing is in practice a more critical problem than condensing, specially when the number of prototypes is not large enough, which occurs very often in many interesting problems.

Wilson’s editing [12] corresponds to the first proposal to edit the NN rule. In a few words, it consists of applying the *k*-NN classifier to estimate the class label of all prototypes in the training set and discard those samples whose class label does not agree with the class associated with the largest number of the *k* neighbors.

The *k*-NCN editing [13] is a way of Wilson’s algorithm particularized for the case of using the *k*-NCN classifier to estimate the class label of prototypes.

Analogously, the GG and the RNG are used for editing the NN classification rule [14]. In brief, after computing the proximity graph (GG or RNG) corresponding to a given training set, we have to discard those prototypes that are misclassified by their graph neighbors.

4.2 Condensing algorithms

Hart’s algorithm [15] is the earliest attempt at minimizing the number of stored prototypes by retaining only a *consistent* subset of the original training set. A consistent subset, say *S*, of a training set *T* is a subset that correctly classifies every prototype in *T* by using the 1-NN rule.

The GG and RNG have also been used to obtain a condensed set of prototypes [16]. This goal is accomplished by retaining only training prototypes with graph neighbors from a different class. As in other condensing schemes, this leads to a subset in which the decision boundaries among classes are close to the ones obtained from the whole training set.

5. THE WEB-BASED LEARNING SYSTEM

The aim of the web-based learning system presented here is to allow the user to carry out a number of interactive experiments with several distance-based algorithms. The system facilitates the comprehension of some techniques studied in a Ph.D. course in Statistical Pattern Recognition, by visualizing the result of applying different algorithms. All experiments have to be on 2-dimensional feature spaces so that the results can graphically be represented. The system focuses on three general topics from the Statistical Pattern Recognition domain: editing, condensing, and classification.

Table 1. Algorithms implemented in the web-based system.

Editing	Condensing	Classification
Wilson’s	Hart’s	*k*-NN
k-NCN	GG	*k*-NCN
GG	RNG	GG
RNG		RNG

In its present form, the learning system consists of a number of web pages describing several algorithms, along with a set of applets to interactively experiment with them. For each one of the general techniques (editing, condensing, and classification), we have initially included the algorithms that appear in Table 1. All web pages have been written in *HTML* and *JavaScript* because they can be run both on Windows and Linux browsers. On the other hand, the different applets have been implemented in *Java* programming language.

The web-based learning system [17] is divided into two frames. As can be seen in Figure 1, the left frame contains a menu to pick a general topic as well as an algorithm. The right frame provides a brief description of the algorithm selected and shows the corresponding applet. For the different topics, one of the main goals is to obtain, as closely as possible, the same appearance for all the applets implemented in the learning system. This makes easier the employment of the algorithms and therefore, users can directly concentrate their efforts on the Pattern Recognition techniques.

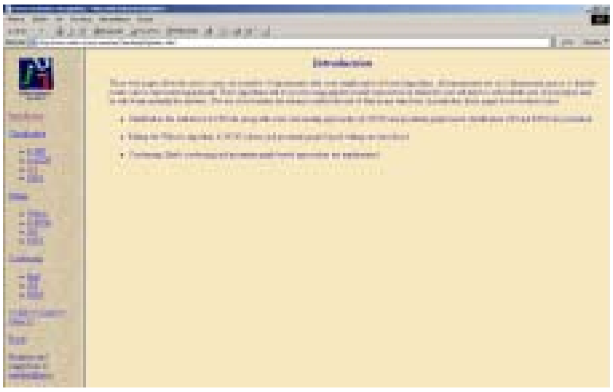

Figure 1. Overview of the web-based learning system.

As already mentioned, the learning system currently focuses on three main topics: classification, editing, and condensing. Within the context of classification, the schemes implemented are *k*-NN, *k*-NCN, GG, and RNG decision rules. For editing, the user can probe Wilson's, *k*-NCN, GG, and RNG algorithms. On the other hand, in the case of condensing, the user can experiment with Hart's, GG, and RNG techniques.

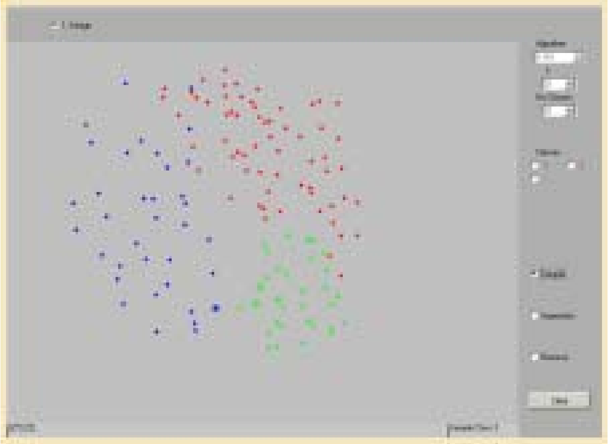

Figure 2. An example for the k-NN classification applet.

Figure 2 shows an example of the *k*-NN classifier applet. It consists of a drawing panel to represent the training prototypes and the sample to be classified, a control panel to define the number of classes and the required parameters (e.g., the *k* value in *k*-NN and *k*-NCN rules), and a status bar to show information (e.g., the coordinates of a point). The user has to define the number of classes and then represent all training prototypes by selecting one of the radio buttons (there is one per class) and clicking on the drawing panel. After defining the training set, the user can insert a new unlabeled sample (represented as a non-filled square) and the system assigns it to the class with a majority of votes among its neighbors. The sample is drawn in the same color as that of the class assigned. The neighbors are represented by a circle.

Figure 3 illustrates an intuitive example for Wilson's editing. More specifically, it shows the design phase in which the user has defined twelve classes and has plotted a number of training prototypes belonging to class 3 (soft diamonds) and class 10 (black diamonds). Since this editing algorithm requires the parameter *k* (neighborhood size), this must be also defined in the design panel. As can be seen, there exists some overlapping between the class regions: some prototypes have been mislabeled.

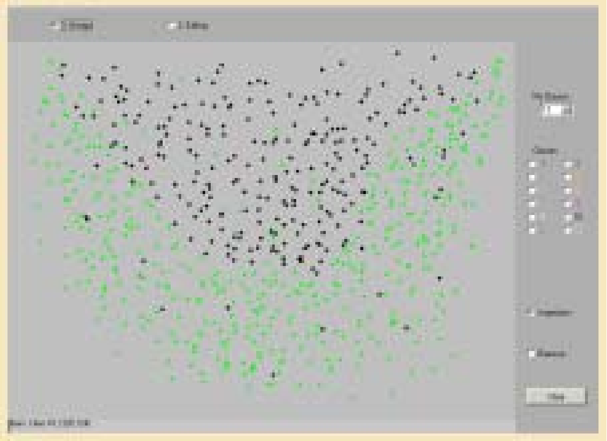

Figure 3. The design phase in Wilson's editing applet.

Figure 4 shows the result of applying Wilson's editing to the training set defined in the previous figure. As expected, the mislabeled prototypes have been removed from the original training set according to the editing algorithm picked.

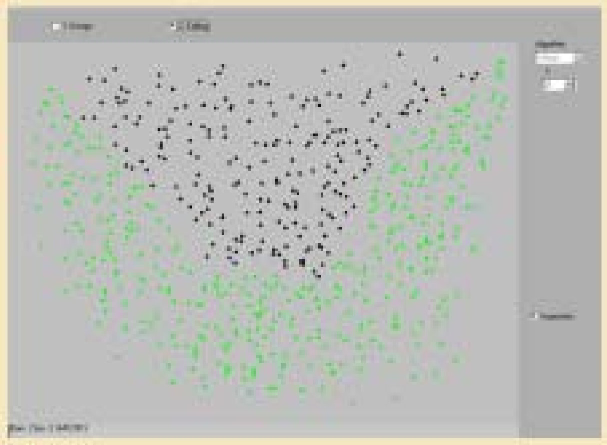

Figure 4. The editing phase in Wilson's applet.

Analogously, Figure 5 illustrates the design phase of a training set with three different classes for the Hart's condensing applet. As already explained in Section 4, in this case, the aim is to obtain a reduced subset of training prototypes that leads to approximately the same performance than the 1-NN rule using the whole set. Thus, as drawn in the present example, it is expected that the training set has no overlapping among different class regions.

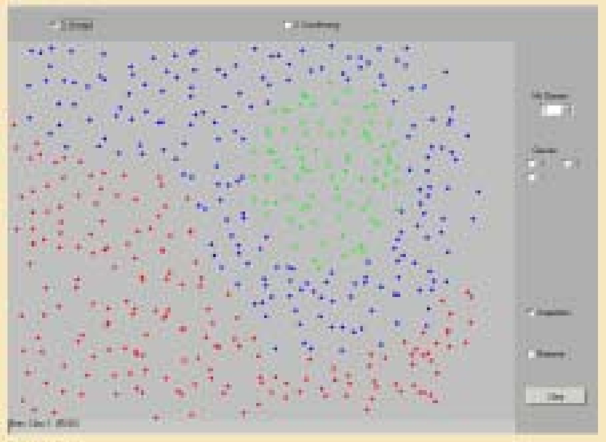

Figure 5. The design phase in Hart's condensing applet.

The result of applying the Hart's algorithm to the training set plotted in the previous figure is provided in Figure 6. As can be seen, the only prototypes that have not been eliminated from the original training set are those defining the decision boundaries.

On the other hand, the user can experiment with all these classification, editing, and condensing algorithms in a combined way, that is, the user can define a synthetic problem (number of classes and training prototypes), edit the training set in order to discard mislabeled prototypes, afterwards apply condensing to the edited set in order to reduce the number of prototypes and finally, classify a set of unlabeled samples by using the condensed set. At each step, the user can probe with different algorithms in order to compare their behavior. This utility allows a more realistic application of these techniques to a classification problem.

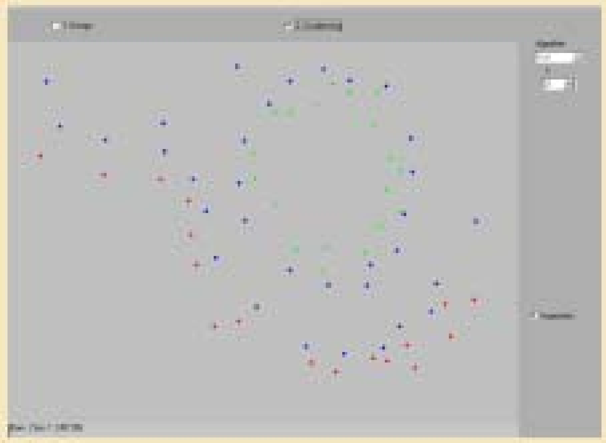

Figure 6. The result of Hart's condensing.

Figure 7 shows the applet corresponding to the combined use of editing, condensing and classification algorithms. It consists of four steps: the design of the training set, the possibility of editing such a set, the possibility of condensing the edited set, and the classification of a number of unlabeled samples. At each step, the user has to specify the particular algorithm to experiment with, along the required parameters.

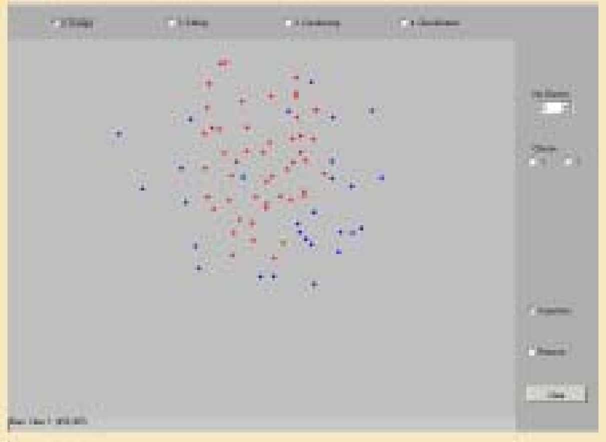

Figure 7. Combined use of editing, condensing and classification algorithms.

As illustrated in Figure 8, now it is possible to classify a set of unlabeled samples, instead of only one as in the case represented in Figure 2.

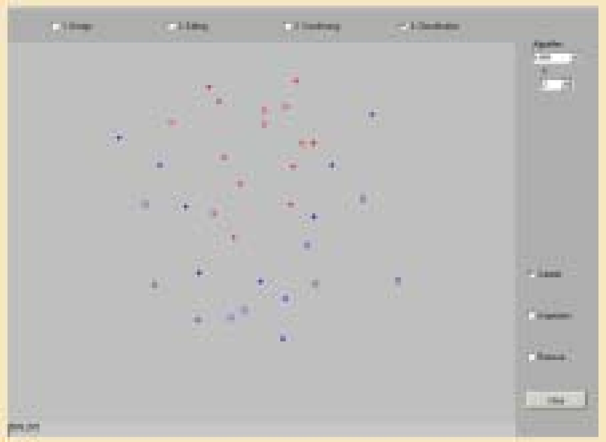

Figure 8. Classification in the "combined" applet.

6. CONCLUSIONS AND FUTURE WORK

The present paper describes a web-based learning system for Statistical Pattern Recognition. More specifically, the system allows the user to experiment with a set of distance-based algorithms related to editing, condensing and classification.

This system is employed as a complementary learning tool for students from a Ph.D. course in Statistical Pattern Recognition at Universitat Jaume I of Castelló. One can interactively probe several classification and prototype selection techniques and also compare their practical behavior on different synthetic 2-dimensional problems. From our experience, the learning system results in a valuable tool so that the students can fully understand the utility of several techniques covered in the aforementioned course.

In its present form, the learning system focuses on distance-based techniques and therefore, a number of different extensions are possible. In particular, our current efforts are addressed to implement other classification algorithms, such as decision trees and Bayesian classifiers. Also, it would be interesting to employ existing data bases as training sets, instead of being interactively defined by the user.

ACKNOWLEDGEMENTS

This work has partially been funded by the Spanish CICYT (Department of Science and Technology) under grant TIC2000-1703-C03-03 and by the Fundació Caixa Castelló-Bancaixa under grant P1-1B2002-07.

REFERENCES

[1] K. Fukunaga, Introduction to Statistical Pattern Recognition. Academic Press, New York, 1990.

[2] R. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches. John Wiley & Sons, New York, 1992.

[3] M. Stefik, Introduction to Knowledge Systems. Morgan Kaufmann, San Francisco, CA, 1995.

[4] Y.H. Pao, Adaptive Pattern Recognition and Neural Networks. Addison-Wesley Publishing Co., Reading, MA, 1989.

[5] F.T. Tschang, T.D. Senta, Access to Knowledge. Pergamon, Oxford, UK, 2000.

[6] R. Duda, P.E. Hart, Pattern Classification and Scene Analysis. John Wiley & Sons, New York, 1973.

[7] B.V. Dasarathy, Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[8] B.B. Chaudhuri, “A new definition of neighbourhood of a point in multi-dimensional space”, Pattern Recognition Letters, vol. 17, pp. 11-17, 1996.

[9] J.W. Jaromczyk, G.T. Toussaint, “Relative neighbourhood graphs and their relatives”, Proc. of IEEE, vol. 80, pp. 1502-1517, 1992.

[10] J.S. Sánchez, F. Pla, F.J. Ferri, “On the use of neighbourhood-based non-parametric classifiers”, Pattern Recognition Letters, vol. 18, pp. 1179-1186, 1997.

[11] P.A. Devijver, J. Kittler, Pattern Recognition: A Statistical Approach, Prentice Hall, 1982.

[12] D.L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data sets”, IEEE Trans. on Systems, Man and Cybernetics, vol. 2, pp. 408-421, 1972.

[13] F.J. Ferri, J.S. Sánchez, F. Pla, “Editing prototypes in the finite sample size case using alternative neighbourhoods”, In: Advances in Pattern Recognition, Lecture Notes in Computer Science 1451, Springer-Verlag, pp. 620-629, 1998.

[14] J.S. Sánchez, F. Pla, F.J. Ferri, “Prototype selection for the nearest neighbour rule through proximity graphs”, Pattern Recognition Letters, vol. 18, pp. 507-513, 1997.

[15] P.E. Hart, “The condensed nearest neighbor rule”, IEEE Trans. on Information Theory, vol. 14, pp. 515-516, 1968.

[16] G.T. Toussaint, B.K. Bhattacharya, R.S. Poulsen, “The application of Voronoi diagrams to nonparametric decision rules”, In: Computer Science and Statistics, Elsevier Science Publishers B.V. North-Holland, pp 97-108, 1985.

[17] http://www.vision.uji.es/~sanchez/Teaching/Alg/