ELSEVIER

Pattern Recognition Letters 18 (1997) 507–513

Pattern Recognition Letters

Prototype selection for the nearest neighbour rule through proximity graphs [1]

J.S. Sánchez [a,*], F. Pla [a], F.J. Ferri [b]

[a] *Dept. d'Informàtica, University Jaume I, E-12071 Castelló, Spain*

[b] *Dept. d'Informàtica i Electrónica, University of València, E-46100 Burjassot (València), Spain*

Received 21 November 1996; revised 23 May 1997

Abstract

In this paper, the Gabriel and Relative Neighbourhood graphs are used to select a suitable subset of prototypes for the Nearest Neighbour rule. Experiments and results are reported showing the effectiveness of the method and comparing its performance to those obtained by classical techniques. © 1997 Elsevier Science B.V.

Keywords: Nearest neighbour; Prototype selection; Gabriel graph; Relative neighbourhood graph

1. Introduction

The Nearest Neighbour (NN) rule has been in practice one of the most widely used non-parametric classifiers. Apart from other advantages common to most non-parametric approaches, the NN rule and its extension to k neighbours (or k-NN rule, in which the k closest neighbours "vote" for the label of the sample) combine their conceptual simplicity with the fact that their asymptotic error rate is conveniently bounded in terms of the optimal Bayes error.

However, the NN rules also present some drawbacks. First, the number of prototypes is usually not large enough to achieve the expected asymptotic performance. Second, the set of prototypes may contain noisy or mislabelled prototypes which usually lead to a decrease in performance.

Prototype Selection (PS) techniques have been proposed as a way of minimizing these problems: they consist of selecting an appropriate reduced subset of prototypes and applying the 1-NN rule using only the selected prototypes. Two different families of PS methods exist in the literature. First, *condensing* aims at selecting a sufficiently small subset of prototypes that leads to approximately the same performance as the 1-NN rule using the whole set. Second, *editing* eliminates erroneous prototypes from the original set and "cleans" possible overlapping among classes, which usually leads to significant improvements in performance.

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 (Devijver and Kittler, 1982) is asymptotically optimal in the sense of Bayes. Nevertheless, when the number of prototypes is not large enough, the classical editing techniques are no longer

* Correspondence author. E-mail: sanchez@inf.uji.es.

[1] Partially supported by projects P1A94-23 (Fundació Caixa-Castelló), GV2110/94 (Generalitat Valenciana) and AGF95-0712-C03-01, TIC95-676-C02-01 (CICYT).

PII S0167-8655(97)00035-4

optimal (Ferri and Vidal, 1992) and some alternative schemes (Kuncheva, 1995) have been proposed to partially overcome this problem.

In this paper the concept of proximity graph (PG) is used to obtain some editing algorithms. This geometric approach arises from previous PG-based condensing algorithms (Toussaint et al., 1985). As in the case of condensing, the editing technique proposed here appears to be a valid alternative to classical schemes in many interesting cases. The comparative results using synthetic and real databases suggest that PG-based editing can be specially suited in the case in which the reference set size is small compared to its intrinsic dimensionality, that is, when classical editing can no longer behave as asymptotically optimal. Finally, some extensions involving combined application of editing and condensing both based on PG are also presented. These combined techniques include a fast recomputation of the corresponding graph structure in a very simple way.

2. Proximity graphs

Let $X = \{x_1, \ldots, x_n\}$ be a set of points in R^d, where n is the number of prototypes and d is the dimensionality of the feature space. A PG, $G = (V,E)$, is an undirected graph with a set of vertices $V = X$, and a set of edges, E, such that $(x_i, x_j) \in E$ if and only if x_i and x_j satisfy some neighbourhood relation. In this case, we say that x_i and x_j are *graph neighbours*. The graph neighbours of a given point constitute its *graph neighbourhood*. The graph neighbourhood of a subset, $S \subseteq V$, consists of the union of all the graph neighbours of every node in S. The PGs used in this work are the *Gabriel Graph* (GG) and the *Relative Neighbourhood Graph* (RNG) (Tüceryan and Chorzempa, 1991; Jaromczyk and Toussaint, 1992).

Let $d(\cdot,\cdot)$ be the Euclidean distance in R^d. The GG is defined as follows:

$$(x_i, x_j) \in E \Leftrightarrow d^2(x_i, x_j) \leqslant d^2(x_i, x_k) + d^2(x_j, x_k), \quad \forall x_k \in X,\ k \neq i, j. \tag{1}$$

In this case, x_i and x_j are said to be *Gabriel neighbours*. In other words, two points are Gabriel neighbours if and only if there is no other point from X laying in the hypersphere centered at their middle point and whose diameter is the distance between them; namely, *hypersphere of influence* of x_i and x_j (Jaromczyk and Toussaint, 1992).

Analogously, the set of edges in the RNG is defined as follows:

$$(x_i, x_j) \in E \Leftrightarrow d(x_i, x_j) \leqslant \max(d(x_i, x_k), d(x_j, x_k)) \quad \forall x_k \in X,\ k \neq i, j. \tag{2}$$

Its corresponding geometric interpretation is based on the concept of *lune* (Jaromczyk and Toussaint, 1992), defined as the disjoint intersection between two hyperspheres centered at x_i and x_j and whose radii are equal to the distance between them. Two points are *relative neighbours* if and only if their lune does not contain other points from X.

The *Minimal Spanning Tree* and *Delaunay Triangulation* (DT), which are also some kind of PG, along with the GG and RNG have been used to solve geometric problems in various domains (Jaromczyk and Toussaint, 1992).

3. Using geometric information in editing

Apart from the DT, the GG and the RNG have also been already used in PS to obtain a condensed set of prototypes (Toussaint et al., 1985). This goal is accomplished by retaining only prototypes with at least one graph neighbour from a different class. As in other condensing algorithms (Hart, 1968), this leads to a reduced subset of prototypes in which the decision boundaries among classes are close to the ones obtained from the whole set.

Condensing algorithms, and PG-based ones in particular, implicitly assume that there are no noisy or mislabelled prototypes in the set. Even more, they assume that there is no overlap among classes. Otherwise, the 1-NN decision boundaries approximated by these algorithms are heavily distorted with regard to the optimal ones given by the Bayes classifier. Unfortunately, the situation just described is the most common in practice and this is the reason why editing is used.

Most classical editing algorithms consist of discarding prototypes with a sufficiently heterogeneous

neighbourhood in terms of labels. PGs can be used as well to obtain an edited set of prototypes. The heuristic approach presented here is also based on the general idea of estimating the label of each prototype from its neighbours but taking into account the concept of graph neighbourhood instead of the Euclidean or other norm-based distance neighbourhood.

Taking into account the definitions given in Section 2, the graph neighbourhood of a point requires that no other point lies inside the union of the zones of influence (i.e., *hypersphere* or *lune of influence*) corresponding to all its graph neighbours. From this neighbourhood relation, we manage to encircle completely a prototype by means of a variable number of neighbours (that is, all its graph neighbours).

It is possible to define classification rules based on the distance induced by the topology in the graph. In particular, k nearest neighbours can be replaced by the graph neighbourhood of a point in the corresponding rule (Sánchez et al., 1997). Consequently, this modified rule can also be used to define new editing procedures to discard prototypes.

The application of PGs to editing has some additional properties with regard to the conventional methods: first, they consider the number of neighbours as a variable feature which depends on every prototype. Second, since the graph neighbourhood of a prototype always tends to surround it, the information extracted from prototypes close to decision boundaries, where uncertainty is highest, may be richer in the sense of the distribution of prototypes.

After computing the graph neighbourhood of every prototype in the input data set, *all* the graph neighbours of a sample (instead of its k nearest neighbours) "vote" for its class. In other words, all prototypes that are surrounding a sample take part in the process of estimating whether it is an outlier or not, regardless of their actual distance to the sample. A first approach to this editing scheme can be expressed in the following way.

1st order graph neighbourhood editing

Step 1. Construct the corresponding PG.

Step 2. Discard those prototypes that are misclassified by their graph neighbours (by the usual voting criterion).

A further refinement of this general idea consists of taking not only the graph neighbours of a point, but also the neighbours of the graph neighbours from its same class (i.e., some of the second level graph neighbours). Actually, what we are trying is to add reliability to the detection of outliers close to the boundaries between classes. Therefore, the previous algorithm can be modified as follows.

2nd order graph neighbourhood editing

Step 1. Construct the corresponding PG.

Step 2. For each sample, p, misclassified by its graph neighbours:

Step 2.1. Consider the subgraph, S, given by p and all its graph neighbours from its same class.

Step 2.2. Discard p if the graph neighbourhood of S has a majority of neighbours from different class than p.

Note that these algorithms do not depend on any critical or tuning parameter, such as number of neighbours or iterations. On the other hand, if we consider the GG and the RNG, we have in principle four different editing algorithms.

4. Combined application of editing and condensing

The algorithms just introduced can be extended in a number of ways. One of the most obvious consists of obtaining an edited-condensed set using the same graph structure. In fact, editing and condensing are two closely related and complementary techniques (Devijver and Kittler, 1982). As mentioned in Section 3, condensing makes sense only when the classes are clustered and well-separated, which constitutes the aim of the editing algorithms.

There exist some practical advantages if we are to apply both editing and condensing using PG approaches. In particular, computation can be saved if part of the proximity information used for editing can be reused for condensing. We propose a simple way to apply PG-based editing-condensing while keeping the computational burden very close to the cost of computing the PG only.

To apply PG-based condensing (Toussaint et al., 1985) after the editing algorithms proposed in Section 3 using the same graph structure, the edges must

be recomputed when the discarded prototypes along with all their edges are removed from the graph. After this step, it may be necessary to add new edges between pairs of non-neighbouring nodes. From the definition of graph neighbours given in Section 2, we can conclude that only pairs of nodes whose zone of influence held an eliminated node can have a new edge. Otherwise, the reason which made a pair of points non-neighbours still holds after editing.

Bearing this in mind, it is possible to store for each pair of non-neighbouring nodes the first detected node inside its zone of influence when the graph was computed for the first time. This makes selection of new edges much faster because only the (few) pairs of nodes whose marked prototype has been discarded during editing need to make a search through the whole set. Note that the nodes are arranged in random order and, therefore, the "first detected node" can be any node. Thus, the algorithm can be written as follows.

Graph neighbourhood editing-condensing

Step 1. Construct the corresponding PG, $G = (V, E)$, and for each pair of points $(p_i, p_j) \in E$ mark the first node that lies inside its zone of influence.

Step 2. PG-based editing.

Step 3. Construct the subgraph, $G' = (V', E')$, corresponding to non-discarded nodes.

Step 4. For each pair p_i, p_j from V', $(p_i, p_j) \notin E'$ and whose stored node is not in V', put an edge if no other node from V' lies inside its zone of influence.

Step 5. PG-based condensing (Toussaint et al., 1985) with the recomputed graph.

It is worth pointing out that modification in Step 1 requires no extra time, while Step 4 is expected to be applied to a reduced number of edges, as confirmed by experimental tests.

5. Experiments and results

Experiments on synthetic and real databases have been performed to illustrate the behaviour of the algorithms presented in this paper. First, classical and proposed editing methods have been compared empirically. Second, different combined editing-condensing schemes are applied to the same problems.

The *Holdout* method averaged over five different random partitions (half for training and half for testing) of each original database, has been used to obtain error rate estimates. Furthermore, the schemes involving internal randomization (e.g. Multiedit) have been repeated five times for each partition. The editing algorithms used in this comparative study are *Wilson's* (Wilson, 1972), *Holdout* (Devijver and Kittler, 1982) and *Multiedit* (Devijver and Kittler, 1982). For condensing, only *Hart's* method (Hart, 1968) has been considered. Typical settings for these algorithms have been tried and the ones leading to the best performance (in the case of Holdout and Multiedit) have been finally included in the figures. The results corresponding to the classification accuracy of the k-NN rule, with increasing values of k (ranging from 1 to 11), using the original reference set have been also included for comparison purposes.

Three different experiments have been included in this section: the first two databases have been taken from a public data repository (Murphy and Aha, 1991) for benchmarking purposes, and the third one from a particular application (Pla et al., 1993). Other experiments have been carried out in this framework (Sánchez et al., 1996), showing a behaviour very similar to that obtained here.

5.1. The Gaussian database

The first experiment consists of a set of seven synthetic databases corresponding to the same problem, but with dimensionality ranging from 2 to 8. There are two classes consisting of multivariate normal distributions with zero mean and standard deviation 1 and 2 in all dimensions, respectively. There are a total of 5,000 patterns: 2,500 in each class.

In the case of editing (Fig. 1), all methods gave very similar performance up to dimension 4 but there is a clear separation in their behaviour as the dimension increases. It can be said that the PG-based edited rules are more insensitive to the ratio between the size of the set of prototypes and its dimensionality. It is worth noting that Multiedit is the best in the lowest dimension as it could be expected from the asymptotical theoretical analysis (Devijver and Kittler, 1982), but its performance degrades from dimension 4 (all prototypes from one of the classes were systematically discarded). The same behaviour is reflected by Hold-

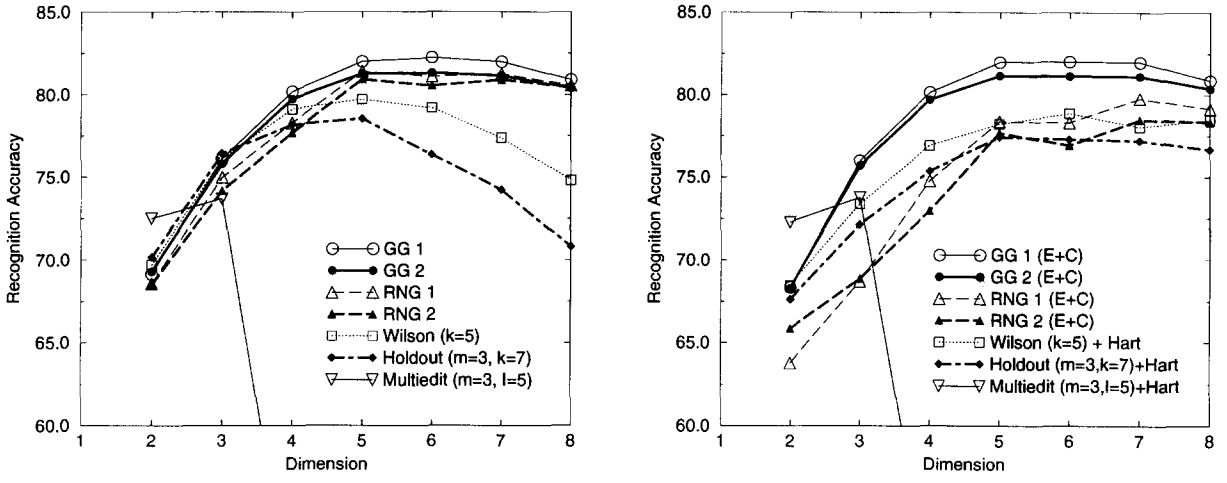

Fig. 1. Comparison of recognition accuracy with varying dimensionalities. (a) Editing. (b) Editing-condensing.

out and Wilson's for higher dimension in a smoother way.

All PG-based edited sets lead to a steadier performance for higher dimensions while keeping similar results for low and moderate dimensionality. The *1st order GG neighbourhood* editing (GG1) is clearly the best for this problem and there is a significant difference between GG-based and RNG-based schemes for dimension 4 and lower. For this experiment, it is worth pointing out that all PG-based editing schemes gave similar results to the best reported ones using non-parametric approaches (Blayo et al., 1995). In addition, these PG-based methods provide better results than the other techniques used in this work.

About editing-condensing in Fig. 1(b), the results are similar to those obtained in editing, although differences are more significant: the highest performance is clearly achieved using the GG.

5.2. The phoneme database

This database was first used in the European ROARS project (Alinat, 1993) for the development of a real time analytical system for French and Spanish speech recognition. The aim of this experiment is to distinguish between two classes representing nasal and oral vowels.

The database contains vowels coming from 1,809 different isolated syllables. The amplitudes of the five first harmonics, normalized by the total energy, are used to characterize each vowel. There are 5,404 patterns: 3,818 for nasal vowels and 1,586 for oral ones.

Even though the PS techniques are not particularly well-suited for this concrete problem (1-NN outperforms any editing scheme), Fig. 2 clearly shows the better results of the PG-based methods compared to the classical ones.

For the edited sets, the highest classification rate is achieved by the *2nd order RNG neighbourhood* editing algorithm (RNG2) and, as in the previous experiment, Multiedit gives the worst performance. About editing-condensing, the GG-based algorithms outperform conventional combinations as well as the RNG-based procedures: this is consistent with the fact that the GG retains more information from the structure of the input data set than the RNG (Jaromczyk and Toussaint, 1992).

5.3. The image database

This data set was already used in (Pla et al., 1993) to study colour segmentation to locate oranges in outdoor scenes under daylight conditions. It consists of values from RGB colour images. There are 19,164 patterns, distributed among three classes (oranges, leaves and sky), with two attributes (the angles of the spher-

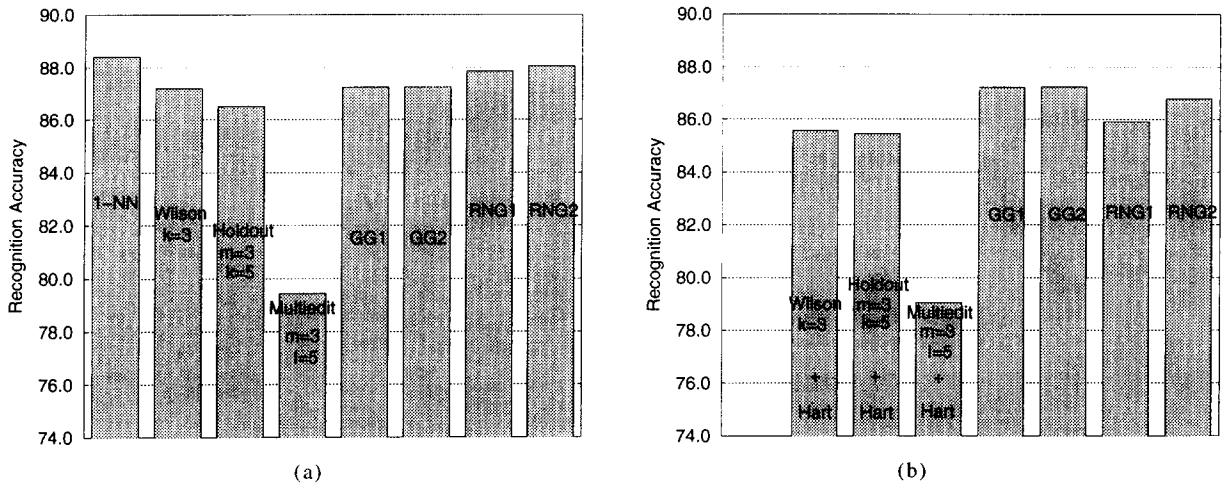

Fig. 2. Accuracy levels for data from an uttered vowel recognition problem. (a) Editing. (b) Editing-condensing.

Table 1
Percentage of remaining prototypes after editing-condensing

	GG 1	GG 2	RNG1	RNG2	Wilson + Hart	Holdout + Hart	Multiedit + Hart
Phoneme	32.23	36.89	17.32	20.21	10.14	10.29	1.04
Image	6.21	6.35	2.94	3.14	0.97	1.86	0.31

ical coordinates of the colour vectors in the RGB space).

For the experiments with this database, we have considered the same format as that used in (Pla et al., 1993) for comparison purposes. Thus, only one unbalanced partition of the input data set (6,386 samples for training and 12,778 for testing) has been made.

As can be seen in Fig. 3(a), all PG-based schemes outperform the classical editing methods, and again the RNG2 gives the highest accuracy (97.95%). It is worth mentioning that performance achieved in (Pla et al., 1993), using a classifier specifically designed for this concrete problem, was close to 99%. From Fig. 3(b), we can notice that the RNG-based combined algorithms obtain systematically worse results than the GG-based ones. This effect can be identified in all experiments in a more or less emphasized way.

6. Concluding remarks

In this paper, the GG and RNG have been used to discard mislabelled prototypes and prototypes belonging to overlapping class regions from the input data set. On the other hand, an algorithm to choose a reduced (edited-condensed) set of prototypes has also been presented. This editing-condensing scheme includes a fast reconstruction of the corresponding graph in a very simple way, keeping the computational burden very close to the cost of only computing the PG.

The number of rejected prototypes by the PG-based and the conventional editing schemes are very similar. The classical methods eliminate about a 1–2% more prototypes than the algorithms proposed here. On the other hand, when applying condensing after editing, the subsets resulting from the PG-based approaches (Toussaint et al., 1985) are about 2–4 times the small number of prototypes retained by Hart's scheme, as can be seen in Table 1.

With respect to editing, the most important fact is that the PG-based methods concentrate their attention on prototypes close to decision boundaries, which gives rise to better classification results than classical schemes in most situations. From a practical point of view, an additional advantage comes from the fact that those schemes deterministically yield edited-

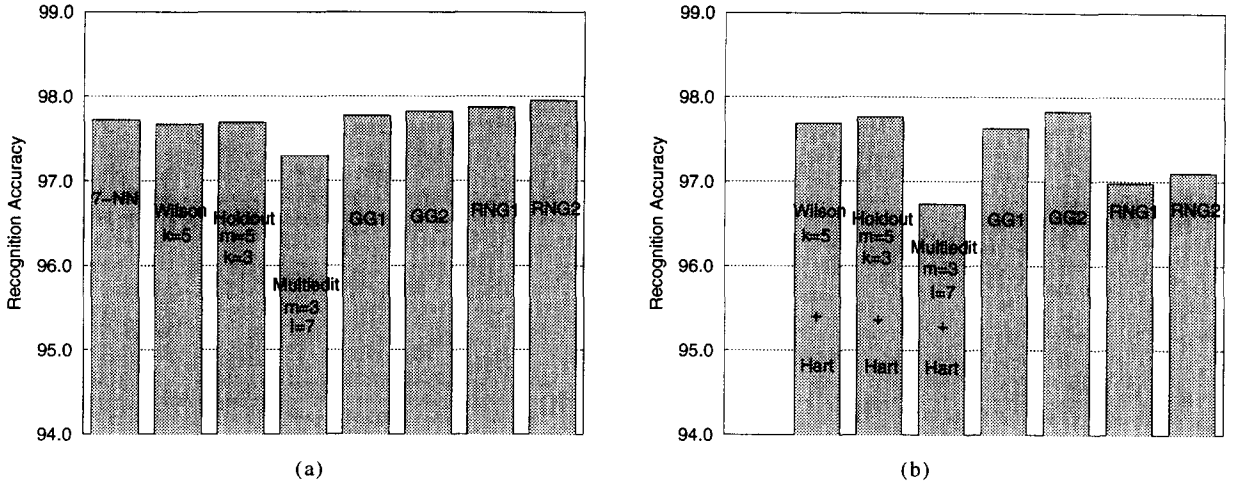

Fig. 3. Classification accuracy for an image recognition problem. (a) Editing. (b) Editing-condensing.

condensed sets independently from any critical tuning parameter.

Currently, further work is directed to improve the algorithms that compute the PG for the first time since the editing schemes proposed in this paper use a heuristic approach (Toussaint et al., 1985) whose expected complexity, for an input data set with n points in d-space, is close to $O(dn^2)$.

References

Alinat, P. (1993). Periodic Progress Report 4: ROARS Project ESPRIT II 5516. Thomson Report.

Blayo, F., Y. Cheneval, A. Guerin-Dugue, R. Chentouf, C. Aviles-Cruz, J. Madrenas, M. Moreno and J.L. Voz (1995). Deliverable R3-B4-P, Task B4: Benchmarks. ESPRIT 6891, Tech. Rept., ELENA: Enhanced Learning for Evolutive Neural Architecture.

Devijver, P.A. and J. Kittler (1982). *Pattern Recognition: A Statistical Approach.* Prentice Hall, Englewood Cliffs, NJ.

Ferri, F.J. and E. Vidal (1992). Small sample size effect in the use of editing techniques. In: *Proc. 11th Internat. Conf. on Pattern Recognition*, 607–610.

Hart, P.E. (1968). The condensed nearest neighbour rule. *IEEE Trans. Inform. Theory* 14, 515–516.

Jaromczyk, J.W. and G.T. Toussaint (1992). Relative neighbourhood graphs and their relatives. *Proc. IEEE* 80, 1502–1517.

Kuncheva, L.I. (1995). Editing for the k-nearest neighbors rule by a genetic algorithm. *Pattern Recognition Letters* 16 (8), 809–814.

Murphy, P.M. and D.W. Aha (1991). UCI repository of machine learning databases. Dept. of Information and Computer Science, University of California, Irvine, CA,

Pla, F., F. Juste, F.J. Ferri and M. Vicens (1993). Colour segmentation based on a light reflection model to locate citrus fruits for robotic harvesting. *Comput. Electronics Agriculture* 9, 53–70.

Sánchez, J.S., F. Pla and F.J. Ferri (1996). Applications of proximity graphs for the nearest neighbour rule. Tech. Rept., Dept. of Computer Science, Univ. Jaume I.

Sánchez, J.S., F. Pla and F.J. Ferri (1997). On the use of neighbourhood-based non-parametric classifiers. In: *Proc. Pattern Recognition in Practice V*, to appear.

Toussaint, G.T., B.K. Bhattacharya and R.S. Poulsen (1985). The application of Voronoi diagrams to nonparametric decision rules. In: *Computer Science and Statistics*. North-Holland, Amsterdam, 97–108.

Tüceryan, M. and T. Chorzempa (1991). Relative sensitivity of a family of closest-point graphs in computer vision applications. *Pattern Recognition* 24, 361–373.

Wilson, D.L. (1972). Asymptotic properties of nearest neighbour rules using edited data. *IEEE Trans. Systems Man Cybernet.* 2, 408–421.