Yugoslav Journal of Operations Research
9 (1999), Number 1, 97-111

PC-PARIS - AN INTERACTIVE SOFTWARE SYSTEM FOR STATISTICAL PATTERN RECOGNITION*

Milan M. MILOSAVLJEVIĆ, Milan Ž. MARKOVIĆ

Institute of Applied Mathematics and Electronics
Kneza Miloša 37, 11000 Belgrade, Yugoslavia

Abstract: Interactive software systems for pattern analysis and recognition represent powerful scientific, researching, and development tools for designing pattern recognition systems. In this paper, we discuss one such software system named PC-PARIS (Pattern Analysis and Recognition Interactive System for PC compatible computers). The variety of pattern recognition and clustering algorithms, implemented in PC-PARIS, are briefly described. As application examples, we firstly present a procedure for the optimal pattern classifier design for the classification of image data using this program. Also, we present an analysis of the sensitivity of statistical pattern classifiers, implemented in PC-PARIS, to the increasing number of features in limited training data set conditions, for synthesized and real data classification cases.

Keywords: Pattern recognition, interactive software system, classifiers, clustering, Bayes error.

1. INTRODUCTION

This paper presents an overview of PC-PARIS, a computer program for designing pattern recognition systems. PC-PARIS stands for PC-based Pattern Analysis & Recognition Interactive System. The aim is to incorporate within a single program the principal algorithms in the field of statistical pattern recognition, thus enabling the user to perform all steps of data analysis under a consistent user interface. Its ultimate purpose is to serve as a development tool for real pattern recognition systems. Thanks to the availability of the source code, it is currently also being used as a powerful tool for intensive research in the field.

* This paper was supported by the Ministry of Science and Technology of the Republic Serbia through the Institute of Mathematics SANU Belgrade.

The need for interactive pattern analysis environments has been recognized for a long time [10], [11]. It arises from the experience that no single method accomplishes the task of achieving low error rate classification regardless of the underlying distribution and the specific data sample; rather, the data have to be subjected to extensive experimental analysis. If the information, thus gathered, is consistent, conclusions about the data structure may be drawn, and a specific method for classification chosen. The aforementioned experimental studies thus represent a key step in the successful system design. This step is greatly facilitated by the use of interactive programs, which hopefully contain a rich library of relevant algorithms. PC-PARIS is designed to serve this purpose.

This paper is divided into 6 sections. Section 2 describes user interaction with the program, and also presents some technical details. In Section 3 we briefly describe the principal data analysis options, except for error estimation, which, due to its importance, is presented in Section 4. Typical applications are the subject of Section 5 while the conclusion is given in Section 6.

2. CONTROLLING PC-PARIS

As its name suggests, PC-PARIS was conceived as an interactive software tool, controlled by the user by sequences of keystrokes from the terminal keyboard, and displaying the results of the data analysis on the screen. PC-PARIS is written entirely in MS FORTRAN PowerStation (version 1.0), a 32-bit compiler for Windows 3.1 (3.11), that creates MS-DOS applications only. In this way, the well-known MS-DOS 640K limitations are overcome and, PC-PARIS, which is a considerable memory consuming software system, can use the entire available operating memory of the given PC station. For example, for the PC-486 DX (4 MB RAM, 33 MHz) workstation, the version of PC-PARIS runs effectively with the maximal number of 10000 training and test data samples, divided into maximally 60 classes with a maximal number of 60 features.

PC-PARIS is a menu-driven program. The desired operation is selected from a choice of possible operations, displayed on the screen, by activating a single key (the appropriate key is displayed along with the title of the corresponding operation). Whenever additional information is needed, it is explicitly requested from the user in the form of a question. In most cases, the default answers to the questions are offered in square brackets. These default answers are selected by simply pressing the Return key, which further simplifies the operation.

PC-PARIS operates in sessions. A session is defined as a sequence of activities between the start and termination of the program. The data to be analyzed are supplied to the system in the form of a set of files. Once read in, the data and the necessary complimentary information are kept in special system files, which are automatically invoked upon the start of the session. Thus, the data have to be defined only once; they then remain in effect in all subsequent sessions, until explicitly changed.

Two different data sets are used in PC-PARIS, called current and training data sets. The current data set corresponds to what is usually called the test data set. For unsupervised learning, like clustering, only one data set is sufficient; other methods use both. The convention adopted here is that each class of training data has to be placed in a separate input file; for the current data set, whose classification is unknown, the placement of patterns in input file(s) is irrelevant. Besides the patterns (vectors) themselves, additional information is also kept in system status files. It includes the number of features, the total number of vectors, the type of distance, the type of data, etc.

PC-PARIS recognizes two types of data: continuous and discrete (actually binary). Some algorithms do not apply for discrete data (e.g., quadratic classifier); as for the others, in some cases special algorithms have been developed to handle high-dimensional discrete data (e.g. k-means clustering).

Each menu consists of a number of options. Some of them, when activated, initiate data processing; others serve to navigate through the menus. A typical menu is shown in Fig. 1.

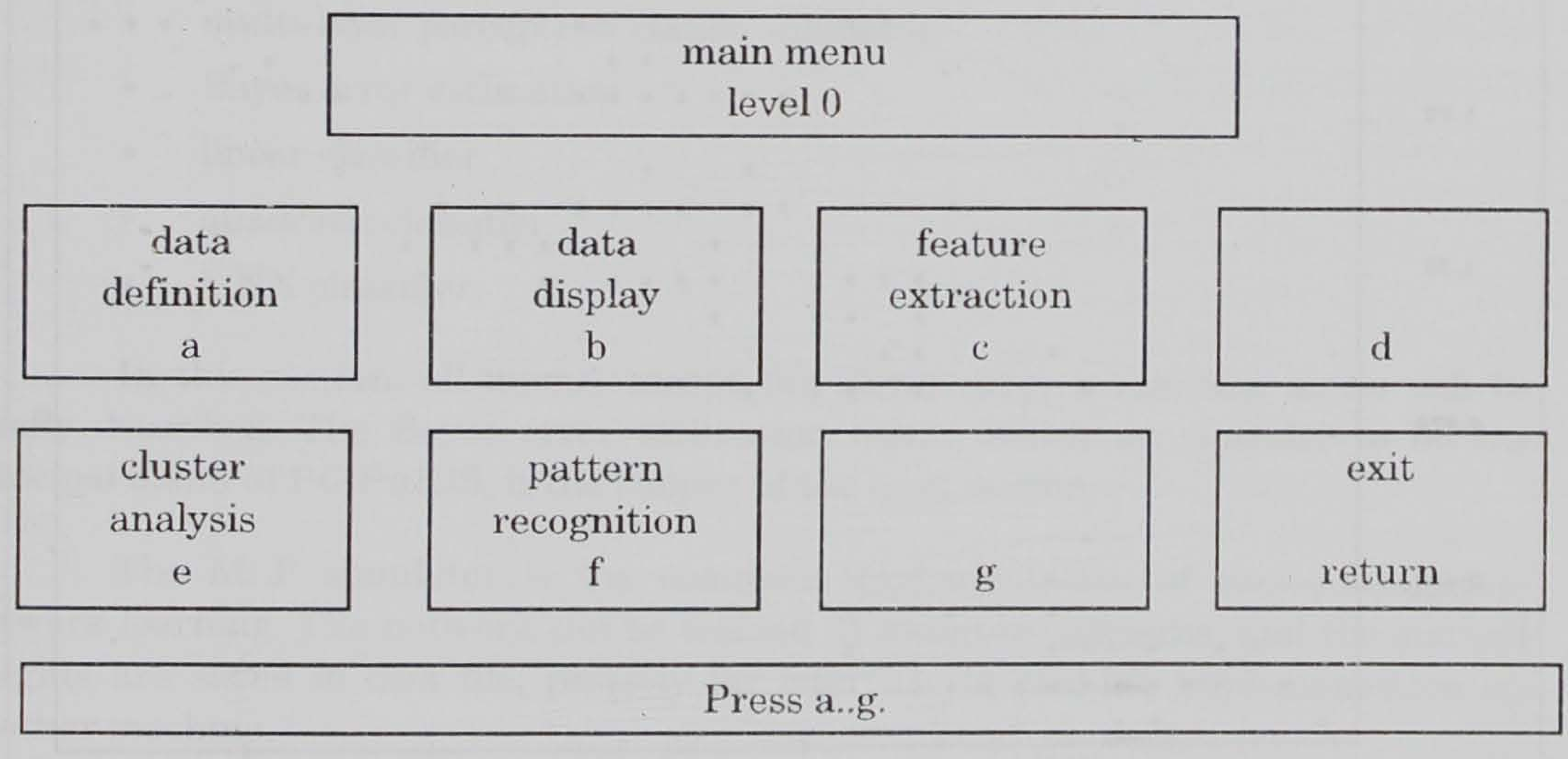


Figure 1: Main menu of PC-PARIS

3. SUBSYSTEMS OF PC-PARIS

The principal subsystems of PC-PARIS are:

- data definition menu
- data display menu
- feature extraction menu
- cluster analysis menu
- pattern recognition menu.

Data definition serves to read data files into PC-PARIS. It also enables the selection of distance (including the Mahalanobis distance), printing of data, and various file operations. Especially noteworthy is the capability to create artificial data sets with given parameters. We found the option for creating two multidimensional normal distributions, with given Bayes error, most useful in our research.

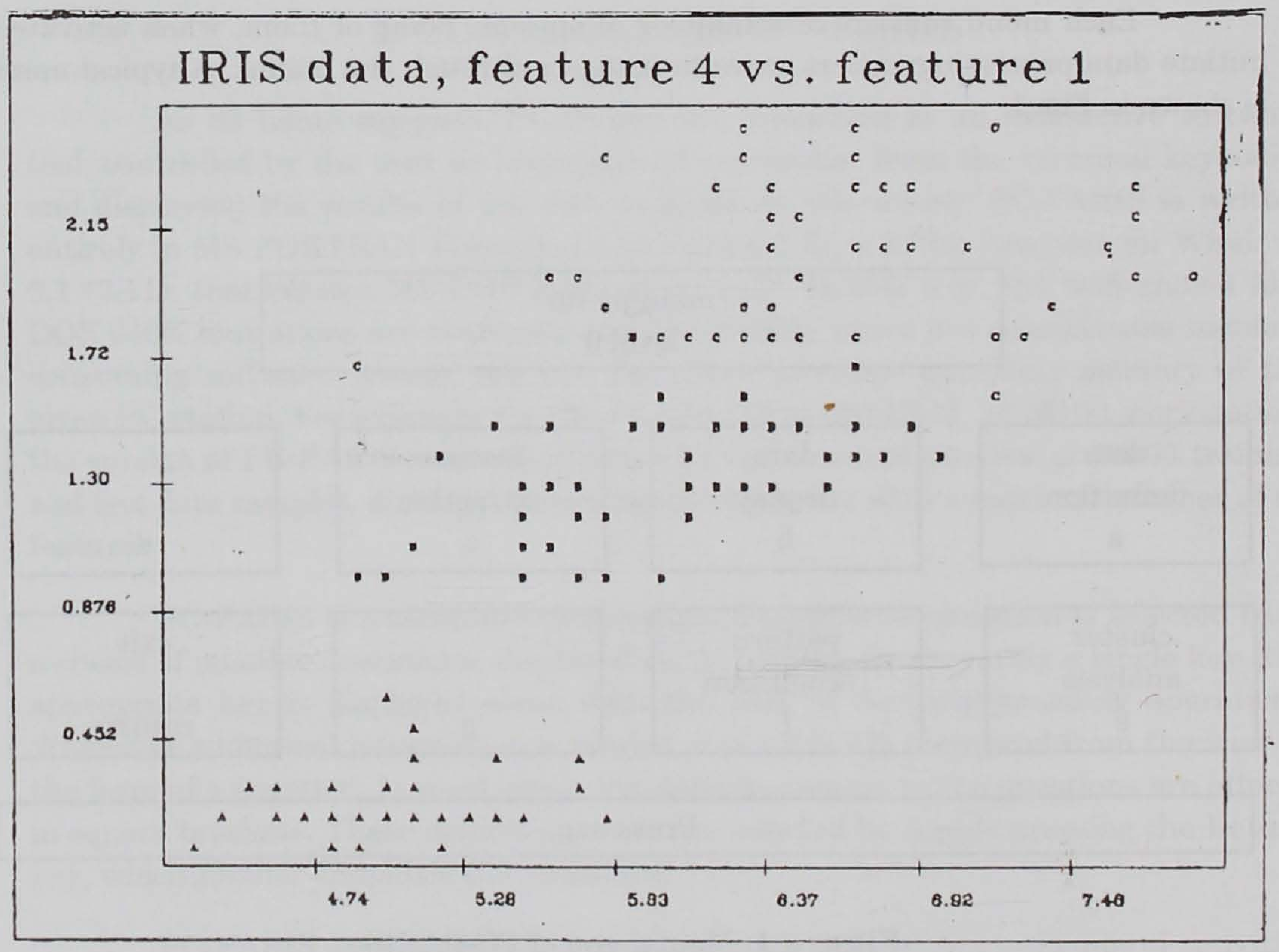


Figure 2: Two-dimensional display of IRIS data

The data display menu displays data along two selected axes. These axes may be selected in the original space, or they may be any of the projections as computed by

Fisher's linear discriminant or the Karhunen-Loeve expansion. Typical data display is shown in Fig. 2.

The menus for clustering include traditional hierarchical methods, like single and complete linkage, Ward's method, median and centroid methods, etc. as well as partitional methods (k-means clustering, ISODATA, and Forgy method). It should be noted at this point that new algorithms have been developed and implemented for the partitional clustering of discrete data, which enable the clustering of binary data of practically unlimited dimensions [3]. Also, the choice of 15 distances (similarity measures) is available for this type of data.

We use the feature extraction menu when we want to reduce the dimensionality of the original data set. This menu includes Fisher's linear discriminant and five versions of the Karhunen-Loeve expansion as described in [4], as well as the proprietary piecewise-linear dimension reduction algorithm, based on an approximation to the sufficient statistics. All these capabilities function in the same way. First, the dimension reducing transformation is computed and stored in disk file(s); second, if desired, the current data set is transformed using the file(s) the lower dimensional space. This new data set automatically becomes the current data set, which means that it is immediately available for further processing.

The pattern recognition menu is used to reach one of the following five submenus:

- multi-layer perceptron (MLP) simulator
- Bayes error estimation
- linear classifier
- quadratic classifier
- k-NN classifier.

In this section, all menus except the Bayes error estimation menu will be briefly described. The Bayes error estimation menu, which we consider to be the principal menu of PC-PARIS, is the subject of the next section.

The MLP simulator is the complete implementation of back-propagation network learning. The network can be trained to recognize patterns, and the learned weights are saved in disk file, possibly for later use in real-life implementation on another machine.

Linear and quadratic classifiers on one hand, and k-NN classifiers on the other, are based on the classical Bayesian classifier theory in the parametric and nonparametric cases, respectively. As in all menus, special care has been given to the ease of use; complete classification results can be obtained with as few as three keystrokes, once the data sets have been defined. The classification results are displayed in the form of a table, such as the one shown in Fig. 3.

CLASSIFICATION RESULTS

Algorithm: quadratic classifier
Number of vectors: 150.

Class	No. of vectors:	List of vectors:									
1	50	1	2	3	4	5	6	7	8	9	10
		11	12	13	14	15	16	17	18	19	20
		21	22	23	24	25	26	27	28	29	30
		31	32	33	34	35	36	37	38	39	40
		41	42	43	44	45	46	47	48	49	50
2	49	51	52	53	54	55	56	57	58	59	60
		61	62	63	64	65	66	67	68	69	70
		72	73	74	75	76	77	78	79	80	81
		82	83	85	86	87	88	89	90	91	92
		93	94	95	96	97	98	99	100	134	
3	51	71	84	101	102	103	104	105	106	107	108
		109	110	111	112	113	114	115	116	117	118
		119	120	121	122	123	124	125	126	127	128
		129	130	131	132	133	135	136	137	138	139
		140	141	142	143	144	145	146	147	148	149
		150									

Figure 3: Classification results in PC-PARIS

4. BAYES ERROR ESTIMATION

In many cases, the ultimate performance measure of a pattern recognition system is the probability of error when Bayes decision theory is employed, i.e. Bayes error. However, this performance indicator is notoriously difficult to compute and/or estimate; therefore, other performance measures have been used to assess the separability power of a set of features. Bayes error is an inherent characteristic of the data set, and does not depend on the classifier that will eventually be employed for classification. It would therefore be highly desirable to be able to estimate it, either for feature selection, feature extraction, or other purposes. We consider that the key feature of PC-PARIS is its menu for Bayes error estimation, which gives the user a relatively faithful picture of the maximum attainable performance at all stages of system design. An early version of the Bayes error estimation facility was reported in [1].

In short, the functioning of the Bayes error estimation menu can be described as follows. In essence, it represents a multi-dimensional, multi-class, non-parametric k-NN error estimation procedure, based on the work of Fukunaga and Hummels [7, 8], and improved in [2]. The output of this method is given in the form of a list of values of functions $e_L^*(k)$ and $e_R^*(k)$, where e^* is the estimated Bayes error, k is the number of neighbors used for estimation, and L and R designate leave-one-out and resubstitution error, respectively. It is shown in [6] that these two curves bound the true Bayes error from above and below, in the case of an unlimited design set. The main contributions of [7, 8] are special procedures for computing these functions, and the finding that no single, theoretically determined value of k gives a reliable estimate. Therefore, in the small sample case, $e_L^*(k)$ and $e_R^*(k)$ have to be plotted against k, and the resultant estimate has to be assessed visually from the plot. Typical plots for Gaussian (a) and non-Gaussian (bimodal) data (b) are shown in Fig. 4. As usual in PC-PARIS, these results are obtained with minimum effort. Once in the Bayes error estimation menu, the user is required to:

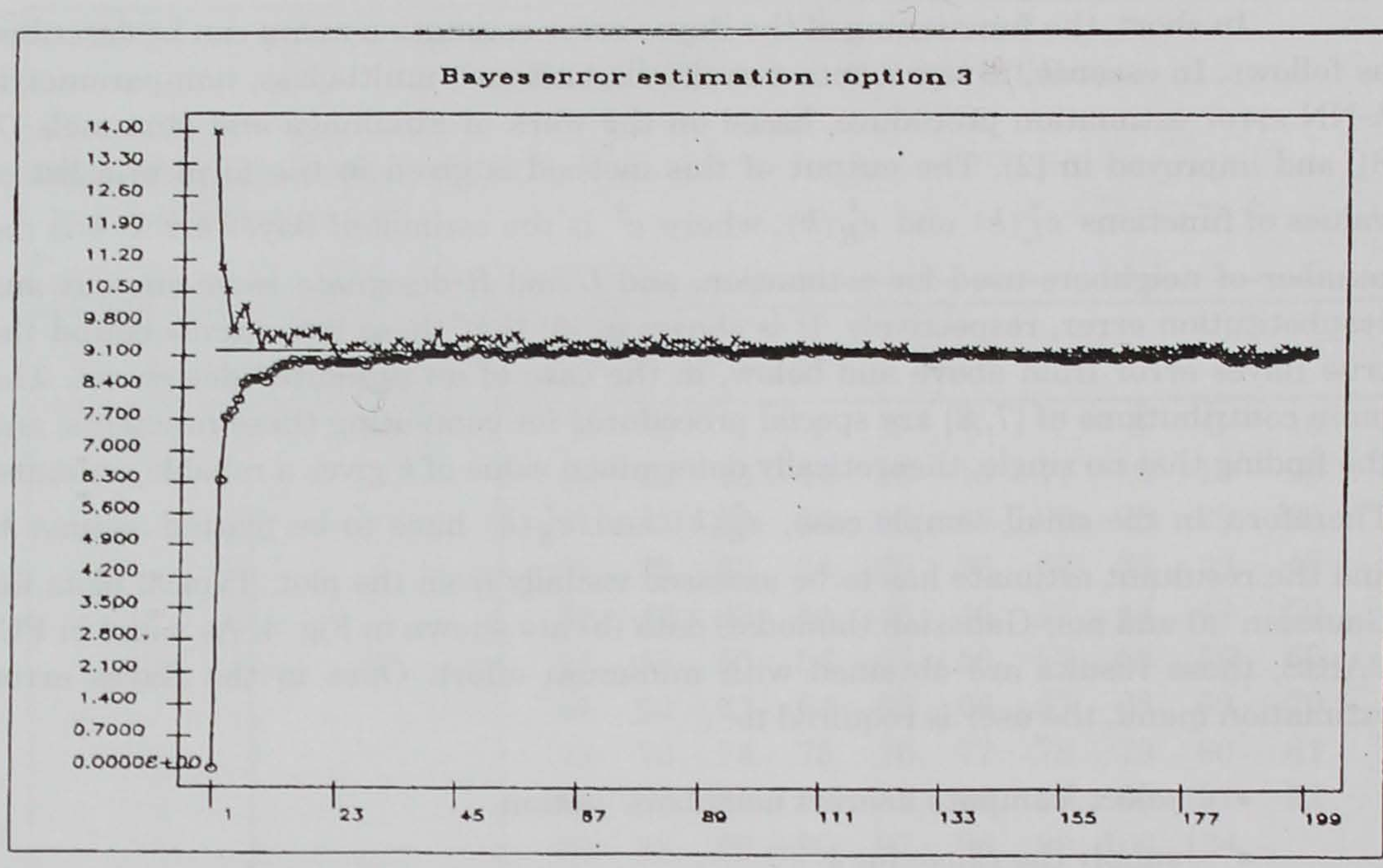
- select "compute nearest neighbors" option
- supply the range for k
- select one or more of the four available estimation options
- activate the plotting program.

The first of the four estimation options mentioned uses the theoretical value for the decision threshold, which gives an unreliable estimate of Bayes error. In contrast, the other three use special optimization procedures, developed in [7, 8], for the computation of the decision threshold. Effectively, that decision threshold value, which minimizes error estimate, is used as the threshold. In practice, these three options give similar results. In a typical case, this procedure involves only 12 keystrokes, most of which are the Return key. To the best of our knowledge, such a software system for Bayes error estimation, not to mention PC-PARIS' other facilities, is not widely available to the pattern recognition community.

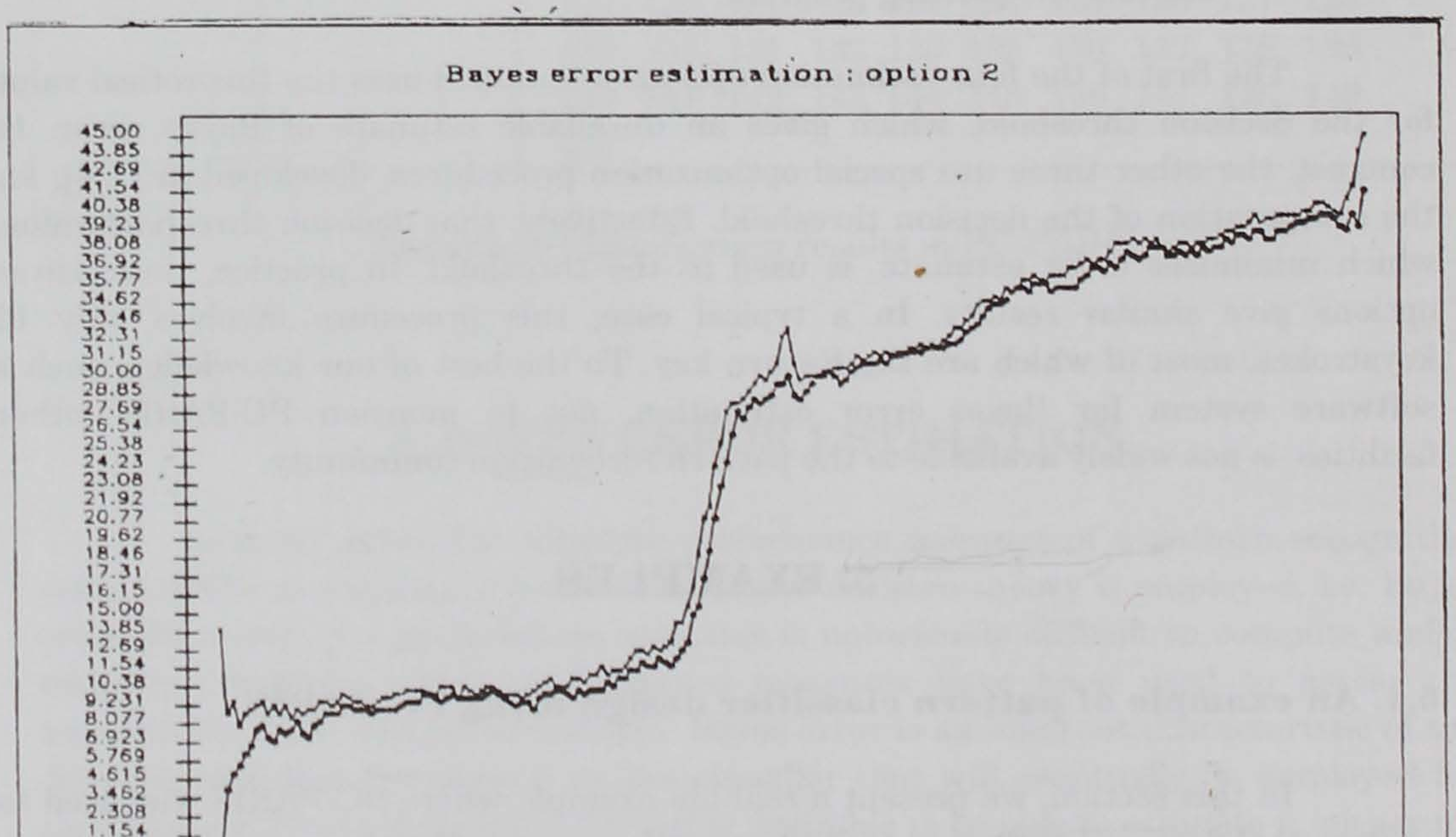
5. EXAMPLES

5.1. An example of pattern classifier design using PC-PARIS

In this section, we present a real-life example where PC-PARIS was used to analyze real data representing two-dimensional contours. The data set consists of five classes of 48 seven dimensional vectors, each vector in a class corresponding to the image of a single object, taken at a different angle. The objects are named T, K, H, TN and AV, and the features represent normalized moments of the digitized image, computed following [5]. The goal is to analyze this data set, and hopefully to design a low error-rate classifier.



(a)



(b)

Figure 4: Bayes error estimation for Gaussian (a) and non-Gaussian (b) data. Upper and lower curves represent leave-one-out and resubstitution errors, respectively

First, we enter the main menu, and then the data definition menu, in order to read data files into PC-PARIS. Since we will eventually do both unsupervised and supervised learning, we will define both data sets (current and training) to be the same, original data set. After this, the data remains in PC-PARIS for all subsequent processing.

Before starting any data analysis, we may want to see selected two-dimensional displays of the raw data. One of them, obtained by simply picking the first and the fourth feature, in the data display menu, is presented in Fig. 5. Not much insight into the data structure is gained this way, so we proceed to the Bayes error estimation menu, to compute the estimate of the error. As already mentioned, the output is given in the form of a plot, see Fig. 6. One may be tempted to conclude from this plot that the Bayes error for this data set is around 9%, and that the performance of any classifier may be close to this figure at best. The shape of the curve also suggests that the deviation from normality is significant. We will, however, defer these conclusions until additional testing is performed.

In the next step, we design linear, pairwise-linear, quadratic and k-NN classifiers, using all of the data set, and test them on the same data. The results obtained are displayed in Table 5.1.1. Note that the Mahalanobis distance was used for k-NN classification. The obtained results are effectively resubstitution errors, which are known to be optimistically biased.

Table 5.1.1: Classifier performance

Classifier	Error
"piecewise" linear	24%
"pairwise" linear	19%
Quadratic	15%
1-NN	12%
2-NN	24%
3-NN	25%

Next, we try to perform dimension reduction. From the main menu, one enters the feature extraction menu, and then Fisher's linear discriminant, piecewise-linear discriminant, or Karhunen-Loeve expansion. We choose, in sequence, all three; for each case, we compute the linear discriminant, transform the samples to four-dimensional space, and finally estimate Bayes error in this space of reduced dimension. The estimation results are presented in Table 5.1.2.

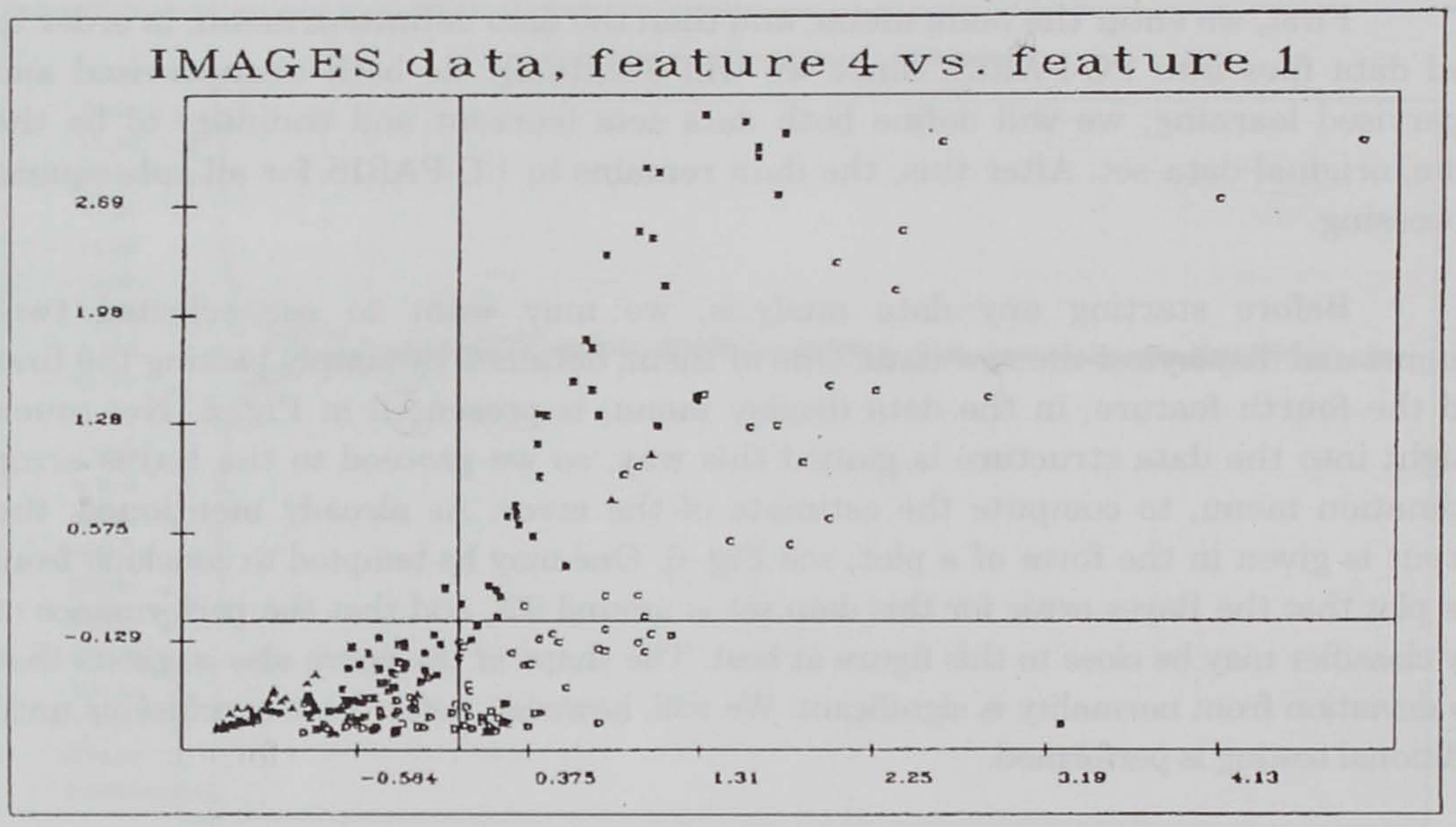


Figure 5: Two-dimensional display of IMAGE data

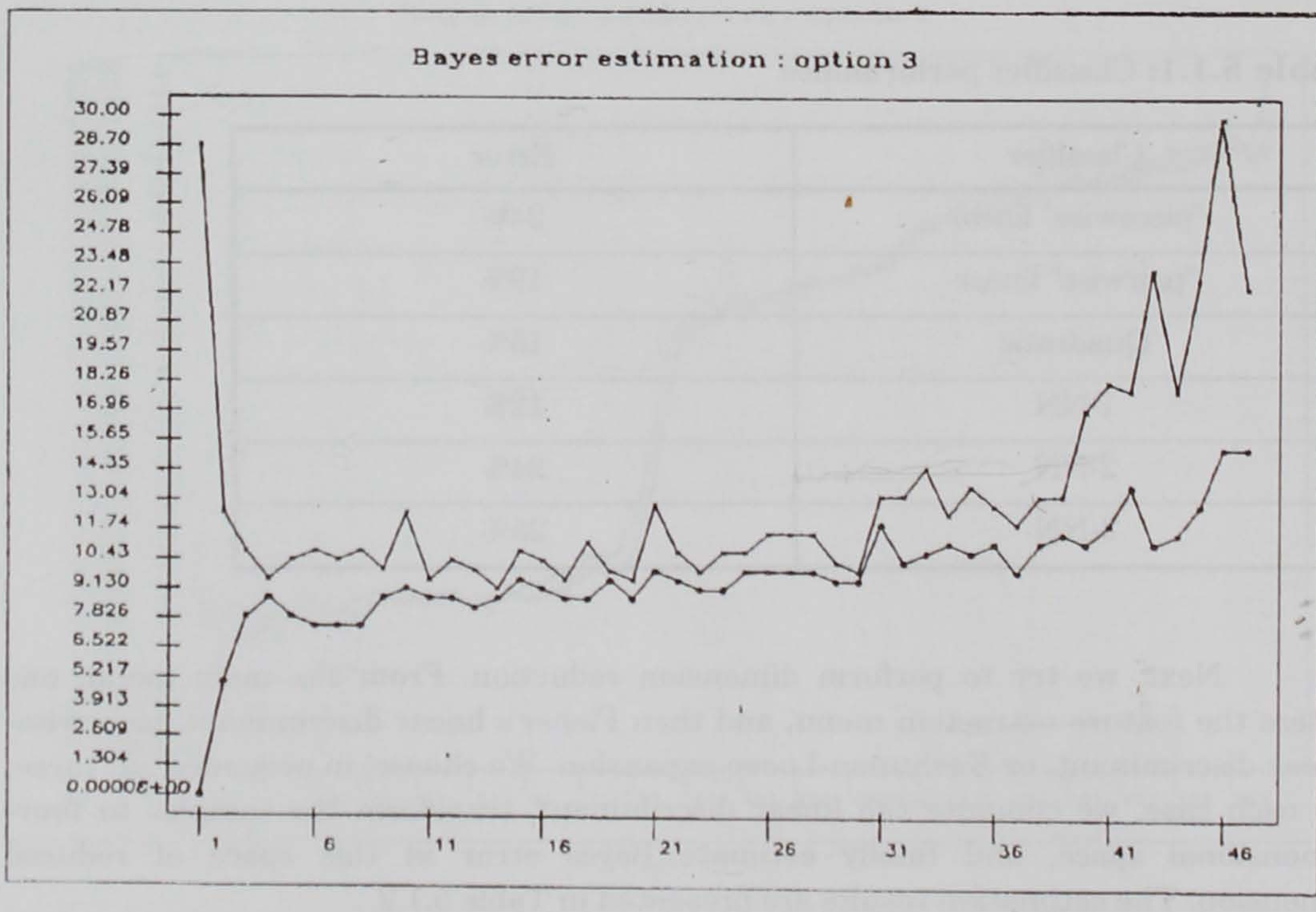


Figure 6: Bayes error estimation of the analyzed data in the original space

Table 5.1.2: Performance of projection algorithms

Algorithm	Bayes error in transformed space
Fisher linear discriminant	14%
piecewise-linear	17%
Karhunen-Loeve expansion	13%

From this table, we conclude that linear transformation to four-dimensional space is inadequate in this case. Instead, we might try to eliminate some of the measurements, and evaluate the discriminating power of the remaining ones. This is easily accomplished by selecting the "select features" option in the data definition menu, which allows us to explicitly pick up a subset of the original set of measurements. Using this capability, and estimating the Bayes error for the remaining six measurements, we obtained the results presented in Table 5.1.3. They indicate that removing feature 5 would not significantly reduce the performance of a carefully designed classifier.

Table 5.1.3: Six-dimensional performance

Omitted feature	Estimated Bayes error in six dimensions
1	20%
2	14%
3	11%
4	13%
5	10%
6	11%
7	12%

By now, we have already collected more evidence for our initial assumption about the data structure. They suggest that the deviation from normality is significant, and that a nonparametric classifier would have to be used in order to get close to the theoretical limit. Finally, we use PC-PARIS to train the multi-layer perceptron (MLP) to classify this data set. These results are summarized in Table 5.1.4.

Table 5.1.4: Classification results using MLP

No. of nodes in hidden layer	Classification error
8	12%
10	11%
12	12%

As already said, our investigations indicate that nonparametric classifiers offer superior performance in this case. A reasonable choice for the classifier would therefore be the multi-layer perceptron, because it is much faster in comparison with the k-NN classifier, especially if implemented in hardware.

5.2. Analysis of statistical pattern classifiers in limited training data set conditions

This section presents a comparative analysis of the limited training data set's influence on the performances of statistical pattern classifiers: *parametric* (two versions of a linear classifier and a quadratic classifier) and *nonparametric* (two versions of k-NN classifiers and a multilayer perceptron). The performances of the k-NN classifiers are considered when using the Mahalanobis distance and for three values of the k nearest neighbors: 3, 5, and 9. As for the MLP, in all of the experiments we fixed the following parameters: $lcoef = 0.01$, initial weights $|w_0| = |0.5|$, using one or two hidden layers with 10 nodes, and 1000 training iterations. The performances of the proposed classifiers are analyzed in accordance with the corresponding Bayes error estimates obtained using a very reliable estimation procedure based on the k-NN approach and the Mahalanobis distance [7, 8] combining the resubstitution and leave-one-out classification error estimation procedure.

Synthesized data: In this paper, standard I-I two-class Gaussian distributed data with Bayes error of 10 %, proposed in [10], were used as synthesized data. To analyze the sensitivity of the proposed classifiers to the increasing number of features in limited training data set conditions, the number of features was chosen to range from 1 to 10 for the same number of $N=100$ vectors for both classes. Even when n changes, the Bayes error stays the same. In Table 5.2.1, the classification errors obtained by the proposed classifiers with mean (E) and standard deviation (σ) are presented. The final Bayes error estimates, in this case, were obtained by averaging the obtained k-NN classification errors for the values of k ranging from 5 to 95. These values were chosen on the basis of the theoretical considerations in [2, 7].

Based on the global results (values E), presented in Table 5.2.1, we can conclude that the smallest classification errors are obtained using the 3-NN "voting" classifier and MLP (2 hidden layers) (3-NN "voting" is a slightly better than MLP (2)). Also, k-NN classifiers show the smallest sensitivity to the increasing number of the features (values σ). Unlike the rest of the considered classifiers, only the MLP clearly shows a decrease in classification error as the number of features increases.

Real data: The real experiments are based on object photographs from five classes, digitized with a frame grabber PC board and Imaging Technology 151 processing system and camera. Forty-seven photos were picked up for every object type. The objects were rotated by arbitrary angles, shifted within the image plane and scaled by camera shift. The resulting gray level images were thresholded to produce binary representation. A boundary follower algorithm is used to identify the boundary contour whose elements were used for one-dimensional (1-D) and two-dimensional (2-D) AR model parameter estimation, as described in [9]. Experimental results obtained by the application of the proposed classifiers in classifying 2-D contours modeled by the mentioned 1-D and 2-D AR models are presented in Tables 5.2.2 and 5.2.3, respectively.

Table 5.2.1: Classification errors (in percents) obtained by applying the proposed classifiers: Synthesized Gaussian I-I data.

Classifier		Feature Number										E	σ
		1	2	3	4	5	6	7	8	9	10		
B. E.		11.86	9.13	6.81	12.76	10.32	7.30	11.09	8.09	12.88	7.33	9.76	2.34
LC (piece.)		14	9.5	7	13.5	11.5	7	12.5	7.5	13.5	7	10.30	3.01
LC (pair.)		14	9.5	7	13.5	11.5	7	12.5	7.5	13.5	7	10.30	3.01
QC		14	9.5	7	14	11	7.5	11	8	12.5	6.5	10.10	2.83
kNN vot.	3	7.5	8.5	4.5	10.5	6.5	7	8.5	6.5	11	6.5	7.70	1.98
	5	11	8	6	12.5	7.5	6.5	8.5	6.5	13	7	8.65	2.58
	9	12.5	10	6.5	13.5	11	7.5	9	9	15	8	10.20	2.76
kNN vol.	3	10.5	9.5	6	12.5	8	6.5	9	7.5	13.5	7	9	2.53
	5	12	8.5	6	13.5	11	8	9.5	9.5	14	8.5	10.05	2.54
	9	14	9.5	7	13.5	12	8.5	11	12	11	9	10.75	2.34
MLP h. l.	1	14	9.5	7.5	12.5	9	6.5	9.5	5.5	8	5.5	8.75	2.81
	2	13.5	9.5	7.5	12	8.5	5.5	8.5	3	7.5	3.5	7.90	3.35

Table 5.2.2: Classification errors obtained by the proposed classifiers: 1-D AR model.

Classifier		Feature Number						E	σ
		1	2	3	4	5	6		
B. E.		13.74	9.60	33.00	22.12	15.34	6.28	16.68	9.640
LC (piece.)		36.17	32.34	35.74	34.89	34.89	25.53	33.26	4.013
LC (pair.)		30.21	21.70	34.47	30.21	25.53	12.34	25.74	7.901
QC		26.38	20.00	37.02	28.09	23.40	17.87	25.46	6.827
kNN vot.	3	12.77	12.77	28.51	31.49	39.57	42.13	27.87	12.726
	5	14.04	16.60	36.60	40.00	46.38	48.94	33.76	14.967
	9	21.70	26.81	42.98	45.53	51.91	51.49	40.07	12.823
kNN vol.	3	11.49	8.51	29.36	20.00	17.87	17.87	17.52	7.268
	5	12.34	11.91	33.62	26.38	24.26	25.53	22.34	8.558
	9	13.19	15.32	34.47	29.79	28.51	28.09	24.90	8.574
MLP h.l.	1	80	58.72	40.43	40	37.45	41.70	49.72	16.695
	2	76.6	61.28	55.74	60.85	42.98	43.40	56.81	12.651

Table 5.2.3: Classification errors obtained by the proposed classifiers: 2-D AR model.

Classifier		Feature Number								E	s
		4	6	8	10	12	14	16	18		
B. E.		2.85	7.90	12.66	14.35	13.55	9.47	7.63	6.02	9.30	4.000
LC (piece.)		30.64	37.87	28.94	31.91	28.51	24.68	25.53	25.11	29.15	4.404
LC (pair.)		27.66	36.17	23.83	21.70	17.02	17.87	17.02	13.62	21.86	7.300
QC		39.57	30.64	20.85	13.62	14.89	11.91	7.66	7.66	18.35	11.41
kNN vot.	3	2.13	22.13	22.13	22.13	22.98	19.57	14.47	18.02	17.94	6.997
	5	2.55	24.68	23.83	25.96	24.68	20.43	17.45	19.57	19.89	7.609
	9	5.11	28.51	24.68	27.23	25.53	22.55	14.47	17.02	20.64	7.943
kNN vol.	3	1.28	12.77	16.60	19.57	17.87	14.47	11.91	10.64	13.14	5.682
	5	1.70	14.04	21.70	22.98	19.57	17.45	14.47	11.49	15.42	6.808
	9	3.83	17.45	26.38	25.11	21.70	17.87	14.47	11.49	17.29	7.429
MLP h.l.	1	70.64	40.85	34.47	37.87	34.47	34.47	30.21	29.79	39.10	13.25
	2	71.49	45.96	40.43	37.02	33.19	35.32	33.19	31.06	40.96	13.22

Regarding real data experiments, based on the results presented in Tables 5.2.2 and 5.2.3, we can conclude that the smallest classification errors are obtained by the 3-NN "volumetric" classifier with the best results using AR models of the first order which correspond to the results presented in [9]. On the other hand, the "piecewise" linear classifier shows the lowest sensitivity to the increasing number of features. In this multi-class case, the MLP shows very bad results suggesting that 1000 training iterations are not enough for the classification of more complicated data than in the Gaussian I-I data case.

6. CONCLUSION

In this paper, the developing possibilities of the interactive software system named PC-PARIS for data analysis and recognition are described. PC-PARIS incorporates a variety of pattern recognition and clustering algorithms (both standard and original algorithms) and is completely open for the inclusion of new algorithms and methods. The subsystems of PC-PARIS are briefly described with special emphasis on the Bayes error estimation subsystem. As application examples of PC-PARIS, we presented a classifier design for two-dimensional contour classification and a comparative experimental analysis of limited training data set influence on the performances of statistical parametric ("piecewise" and "pairwise" linear and quadratic) classifiers and nonparametric ("voting" and "volumetric" k-NN and multilayer perceptron) classifiers. Based on the presented material, we can conclude that PC-PARIS presents a powerful software tool for the research and development of pattern recognition systems.

REFERENCES

[1] Buturović, Lj.J., "An interactive software package for nonparametric Bayes error estimation", *Proc. of Second Conference of the International Federation of Classification Societies*, Charlottesville, VA, June 27-30, 1989.

[2] Buturović, Lj.J., and Marković, M.Ž., "Improving k-nearest neighbor Bayes error estimates", *Proc. of 11th IAPR International Conference on Pattern Recognition*, The Hague, The Netherlands, August 30-September 3, 1992.

[3] Buturović, Lj.J., and Veinović, M.Đ., "Clustering of binary data using the ISODATA algorithm", *Proc. of Second Conference of the International Federation of Classification Societies*, Charlottesville, VA, June 27-30, 1989.

[4] Devijver, P.A., and Kittler, J., *Pattern Recognition: A Statistical Approach,* Englewood Cliffs, Prentice/Hall International, 1982.

[5] Dudani, S., Breeding, K., and McGhee, R., "Aircraft identification by moment invariants", *IEEE Trans. Computers*, 26 (1977) 39-45.

[6] Fukunaga, K., "The estimation of the Bayes error by the k-nearest neighbor approach", in: L.N. Kanal, and A. Rosenfeld (eds.), *Progress in Pattern Recognition 2*, Amsterdam, North-Holland, 1985, 169-187.

[7] Fukunaga, K., *Introduction to Statistical Pattern Recognition*, Academic Press INC., Harcourt Brace Jovanovich, Publishers, Boston-San Diego-New York-London-Sydney-Tokyo-Toronto, 1990.

[8] Fukunaga, K., and Hummels, D.M., "Bayes error estimation using Parzen and k-NN procedures", *IEEE Trans. Pattern Anal. Mach. Intell.*, 9 (1987) 634-643.

[9] Pejnović, P., Stanković, S., and Delevski, V., "AR modeling and the classification of 2-D shapes", (in Serbian), *Zbornik Radova XXXVII Konf. ETAN*, Beograd, 20-23 September 1993., XIII - VI, ES, 45-50.

[10] Sammon, J.W., "Interactive pattern analysis and classification", *IEEE Trans. Computers*, 19 (1970) 594-616.

[11] Voss, K., Hufnagl, P., and Klette, R., "Interactive software systems for computer vision", in: L.N. Kanal, and A. Rosenfeld (eds.), *Progress in Pattern Recognition 2*, Amsterdam, North-Holland, 1985, 57-78.