The *Automated* Multidimensional Detective

Alfred Inselberg* & Tova Avidan
Computer Science Department
Tel Aviv University, Israel
aiisreal@math.tau.ac.il

Abstract

Automation has arrived to Parallel Coordinates. A geometrically motivated **classifier** is presented and applied, with both training and testing stages, to 3 real datasets. Our results compared to those from 23 other classifiers have the least error. The algorithm is based on parallel coordinates and :

- has very low computational complexity in the number of variables and the size of the dataset – contrasted with the very high or unknown (often unstated) complexity of other classifiers,
- the low complexity enables the rule derivation to be done in near real-time hence making the classification **adaptive** to changing conditions,
- provides comprehensible and explicit rules – contrasted to neural networks which are "black boxes",
- does dimensionality selection – where the minimal set of *original* variables (not transformed new variables as in Principal Component Analysis) required to state the rule is found,
- orders these variables so as to optimize the clarity of separation between the designated set and its complement – this solves the pesky "ordering problem" in parallel coordinates.

The algorithm is **display independent**, hence it can be applied to very large in size and number of variables datasets. Though it is instructive to present the results visually, the input size is no longer display-limited as for *visual* data mining.

Motivation and the Algorithm

The display of multivariate datasets in parallel coordinates (abbr. ||-coords) transforms the search for relations into a 2-D pattern recognition problem. Until now the discovery involved a skillful interaction between the "detective" and the data display; a process which was illustrated in the "Multidimensional Detective" [3]. It is not surprising that

*Senior Fellow San Diego SuperComputing Center, and Multidimensional Graphs Ltd, Raanana, Israel

For Color Plate, See Page 151.

0-7695-0431-0/99 $10.00 © 1999 IEEE

the most persistent requests and admonitions were for tools which, at least partially, automate the Knowledge Discovery process.

Classification is a basic task in data analysis and pattern recognition and an algorithm accomplishing it is called a **Classifier** or **Rule Finder** [2], [6], [7]. The input is a dataset P and a designated subset S. The output is a characterization, that is a set of conditions or rules, to distinguish elements of S from all other members of P the "global" dataset. The output may also be that there is insufficient information in the dataset to provide the desired distinction. As an example, a bank manager having data on all his customers (here this is the set P) may want a rule to distinguish the most profitable (or riskiest) customers (this would be the set S) from all others. Such a task, assuming that there is sufficient information in the dataset, can be assigned to a classifier. This paper consisting of the description and results from an automatic classifier based on ||-coords is in a sense the sequel to [3].

With parallel coordinates a dataset P with N variables is transformed into a set of points in N-dimensional space. In this setting, the designated subset S can be described by means of a hypersurface which encloses just the points of S. In practical situations the strict enclosure requirement is dropped and some points of S may be omitted (in the lingo of Data Mining these points are called "false negatives"), and some points of $P - S$ are allowed (these are the "false positives") in the hypersurface. The description of such a hypersurface is equivalent to the rule for identifying, within some acceptable error, the elements of S. This is the *geometrical* basis for the classifier presented here. The algorithm accomplishing this entails:

- use of an efficient "wrapping" algorithm to enclose the points of S in a hypersurface S_1 containing S and typically also some points of $P - S$; so $S \subset S_1$, of course such an S_1 is not unique [1],
- the points in $(P - S) \cap S_1$ are isolated and the wrapping algorithm is applied to enclose them, and usually also a few points of S_1, producing a new hypersurface S_2 with $S \supset (S_1 - S_2)$,
- the points in S not included in $S_1 - S_2$ are next marked for input to the wrapping algorithm, a new hypersurface S_3 is produced containing these points as well as some other points in $P - (S_1 - S_2)$ resulting in $S \subset (S_1 - S_2) \cup S_3$,
- the process is repeated alternatively producing upper and lower containment bounds for S; termination occurs when an error criterion (which can be user specified) is satisfied or when convergence is not achieved.

Basically, the "wrapping" algorithm is a fast way of producing a hypersurface enclosing tightly a given point set. The kind of surface produced is a convex-hull approximation. The efficiency of the version implemented here is due to the use of the ||-coords representations of N-dimensional objects applied in the description of the resulting hypersurface [4]. To summarize, initially the wrapping S_1 encloses all the points of $S = S_0$. Then in the attempt to remove all extraneous points a *cavity* is created by the subsequent wrapping. Such cavities are generically denoted by S_{2n} for $n = 1, 2, ...$ Usually some of the points of S are enclosed in S_{2n}, so a correction follows with a S_{2n+1}, the hypersurfaces with odd subscript, which enclose and add these points to the previous approximation for the enclosure of S. Such a correction may also add some points of $P - S$ which need to subsequently removed, or better reduced, to provide an increasingly tighter bound. So the process entails bounding the designated set S alternately from above and below providing, in case of convergence,

[1] To avoid unnecessary verbiage by a statement $S_j \subset S_k$ we also mean that the set of points enclosed in the hypersurface S_j is contained in the set of points enclosed by the hypersurface S_k.

an increasingly better approximation for S. It can and does happen that the process does not converge when P does not contain sufficient information to characterize S. It may also happen that S is so "porous" (i.e. sponge-like) that an inordinate number of iterations are required.

At step r the output is the description of the set S_r which consists of:

- a list of the minimum number of variables needed to describe S *without loss of information.* Unlike other methods, like the Principal Component Analysis (PCA), the classifier discards only the redundant variables. It is important to clarify this point. A subset S of a multidimensional set P is not necessarily of the same dimensionality as P. So the classifier finds the dimensionality of S in terms of the original variables and retains only those describe S. That is, it finds the *basis* in the mathematical sense of the smallest subspace containing S, or more precisely the current approximation for it. This basis is the minimal set M_r of variables needed to describe S. We call this dimensionality **selection** to distinguish it from dimensionality *reduction* which is usually done *with* loss of information. Retaining the original variables is important in the applications where the domain experts have developed intuition about the variables they measure. The classifier presents M_r *ordered according to a criterion which optimizes the clarity of separation.* This may be appreciated with the example provided in the attached figure, in addition,
- the current approximation of the rule stated in terms of conditions on the variables M_r, which constitutes the description of the current hypersurface, is obtained.

So on convergence, say at step $2n$, the description of S is provided as :

$$S \approx (S_1 - S_2) \cup (S_3 - S_4) \cup ... \cup (S_{2n-1} - S_{2n})$$

this being the terminating expression resulting from the algorithm.

The implementation allows the user to select a subset of the available variables and restrict the rule generation to these variables. In certain applications, as in process control, not all variables can be controlled and hence it would be useful to have a rule involving such variables that are "accessible" in a meaningful way. There are also two options available :

- either minimize the number of variables used in the rule, or
- minimize the number of steps, in terms of the unions and (relative) complements, in the rule.

In the first case, when the first hypersurface S_1 is found, the variables occurring in its description are the minimum number of variables needed to describe S. From this point on the algorithm can be restricted to use only these. If convergence is achieved a rule involving this minimal set of variables is obtained; we fondly refer to this variation as **Enclosed Cavities** and abbreviate it by **EC**. By contrast, when the algorithm is allowed to operate on all the initially selected variables **at each step**, the number of operations in the terminating expression is reduced. This variation of the classifier is called **Nested Cavities** (abbr. **NC**). Clearly the minimal set of variables needed to specify S is not given by **NC**. In practice, the reduction in the number of steps between **EC** and **NC** turns out to be substantial.

It was already pointed out that, dimensionality **selection** involves finding the dimensionality, M, of the subset S in terms of the original variables. To illustrate, let us consider a set P in N-D and a subset $S \subset P$ which is contained in a p-dimensional ($p < N - 1$) plane (called a p-flat) of N-D. This p-flat is positioned so that every one of it's points can be described as a linear combination of p variables with p being the minimum such number (i.e. basis). Then **EC** will return $M = p$. Now S is rotated so that minimally $q > p$ of the original variables are needed to describe every one of the points of S as linear combinations. In this case $p < q = M \leq N$. As an example, if $N = 3$ take P to be a cube with axes parallel edges, and $S \subset P$ a line segment which is parallel to one of the axes so that $M = 1$. The line segment S is now rotated so that it is no longer parallel to any of the axes. In that case $M = 2$ if S is contained in a plane parallel to a principal plane or $M = 3$ if not. This shows that there is still room for dimensionality *reduction* methods like PCA to be applied **after** dimensionality *selection*. For this will result in new variables involving the *minimum* number of original variables. The prospect is certainly worth exploring. In the 3 cases presented next the dimensionality was lowered significantly not only by **EC** but also by **NC** to less than half and in one case to about 1/4 of the original variables.

One of the pesky problems in using parallel coordinates for viewing a specific dataset is to somehow find an axes permutation which is "good" (i.e. provides rich visual cues on what may be true or not) about the **specific** dataset. There is an inherent ordering emerging from dimensionality selection which, as we see below, answers this need well. This ordering is completely dataset specific. Further, since the algorithm is *display independent* there is no inherent limitation as to the size and number of variables in the dataset. The most significant limitation then in visual data mining is finally overcome. The visual aspects can now be used for displaying the result as well as exploring the salient features of the distribution of data brought out by the classifier.

This is not the right forum to analyze the computational complexity and other intricacies of the algorithm. It is worth pointing out that achieving an "optimum", in the sense of minimizing the number of cavities, turns out to be an NP-complete problem. Still the next best thing is done here in terms of discovering the cavities in order of decreasing size. Other relevant aspects are:

- an approximate convex-hull boundary for each cavity is obtained,

- utilizing properties of the representation of multidimensional objects in ||-coords, a very low polynomial worst case complexity of $O(N^2|P|^2)$ in the number of variables N and dataset size $|P|$ is obtained; it is worth contrasting this with the often unknown, or unstated, or very high (even exponential) complexity of other classifiers,

- an intriguing prospect, due to the low complexity, is that the rule can be derived in near real-time making the classifier **adaptive** to changing conditions,

- the minimal subset of variables needed for classification is found,

- the rule is given explicitly in terms of conditions on these variables, in terms of included and excluded intervals, and provides "a picture" showing the complex distributions with regions where there is data and "holes" with no data; that can provide significant insights to the domain experts,

Results

Three datasets, benchmarks in classification, are used to test the classifier. The results are then compared to those obtained with other well-known classifiers.

On the classifiers

During 1990-1993, Michie, Spiegelhalter and Taylor [5], on behalf of the ESPRIT program of the European Union, made extensive studies of several classifiers applied to diverse datasets. About 23 different classifiers were applied to about 20 datasets for comparative trials in the **StatLog** project. This was designed to test classification procedures on large-scale commercially important problems in order to determine suitability of the various techniques to industry. There were three main types of classifiers used:

1. **Extension to Linear Discrimination**

   This group includes algorithms which start with linear combinations and are followed by non-linear transformations of various sorts. Seven (7) different such classifiers were used named : *Discrim, Logdisc, Quadisc, SMART, Backdrop, Cascade and DIPOL92.*

2. **Decision Trees and Rule-Based Methods**

   The 9 decision tree and rule-based methods used were : *NewID, AC^2, Cal5, C4.5, CART, IndCART, Baytree, CN2, ITRule*

3. **Density Estimates**

   These algorithms estimate probability density at each point, they were : *Naive-Bay, CASTLE, ALLOC80, K-NN, RBF, Kohonen and LvQ.*

Data, Results and Comparisons

Satellite image data

Satellite data is used extensively in military, meteorological, earth resources planning and lots of other applications. The specific dataset used in *Statlog* is from a region in Australia. It consists of multi-spectral values and associated classification according to ground type and can be found in the *Statlog* ftp site. Each frame consists of four digital images of the same scene in different spectral bands, two in the visible and two in the near infra-red region. There are 36 variables (the attributes) and the class attribute for six (6) soil types i.e. the six classes to be characterized by the classifier(s). The data has 4435 samples(data items) for the training set and 2000 samples in the test set. By way of explanation, for validation the dataset is partitioned into *training* and *testing* subsets, the "popular" proportions being about 2/3 to 1/3. The rule is derived, by the classifier, on the the training set and tested on the remainder of the data; the error pertains to the false "positives" and "negatives". The important measure is the *test error* which should be as small as possible; to a lesser extent the difference between the train and test errors should be small. The comparative results are shown in Table 1 below.

By way of an example, a rule found by **NC** for one of the classes of this dataset required 23 out of the 35 attributes and was done in 4 iterations (i.e.only 4 hypersurfaces were needed). The class size was $|S| = 479$ (out of a total of 4435 items). It turns out that $|S_1| = 1291$, $|S_2| = 831$, $|S_3| = 162$, $|S_4| = 143$. Notice that,

$$479 = |S| = (460) + (19) = |(S_1 - S_2) \cup (S_3 - S_4)|$$

So in this case the rule found is "exact". Of course, this should not be taken literally since it depends on the actual data items used for training as can be seen from the test error. Not surprisingly the larger the datasets the more reliable are rules found and the closer are the training and testing errors. In a great many cases S_2 turned out to be the hypersurface requiring the largest number of variables for its definition. We conjecture

RANK	CLASSIFIER	Error rate %	
		Train	Test
1	**Nested Cavities (NC)**	**4.3**	**9.0**
2	k-NN	8.9	9.4
3	LVQ	4.8	10.5
4	DIPOL92	5.1	11.1
5	RBF	11.1	12.1
6	ALLOC80	3.6	13.2
7	IndCART	2.3	13.8
8	CART	7.9	13.8
9	Backprop	11.2	13.9
10	Baytree	2.0	14.7
11	CN2	1.0	15.0
12	C4.5	4.0	15.0
13	NewID	6.7	15.0
14	Cal5	12.5	15.1
15	Quadisc	10.6	15.5
16	AC^2		15.7
17	SMART	12.3	15.9
18	Cascade	11.2	16.3
19	Logdisc	11.9	16.3
20	Discrim	14.9	17.1
21	Kohonen	10.1	17.9
22	CASTLE	18.6	19.4
23	NaiveBay	30.8	28.7
24	ITrule	Failed	Failed

Table 1: Summary of the *StatLog* results and comparison with the **Nested Cavities (NC)** classifier for the satellite image data. The error is averaged over the six classes.

that this is an indication of the existence of many "borderline" cases (i.e. close elements in the class S and it's complement) and it may suggest that the class definition may be "fuzzy".

Vowel recognition data

This is an interesting problem in speech recognition where likely users are the physically handicaped, or those with "busy hands", or without keyboard access. The tasks may involve changing radio frequencies in airplane cockpits, asking for stock-market quotations on the telephone etc.

The process involves digital sampling of speech then acoustic signal processing, followed by recognition of the phonemes, groups of phonemes and words. The goal here is a speaker-independent rule based on 10 variables of 11 vowels occurring in various words spoken (recorded and processed) by 15 British male and female speakers. Deterding [1] collected this dataset of vowels and which can be found in the CMU benchmark repository on the WWW. There are 528 entries for training and 462 for testing. Three other types of classifiers were also applied to this dataset: neural networks and k-NN by Robinson & Fallside [8], and Decision trees by Shang and Breiman [9]. For the sake of variety both versions of our classifier were used and a somewhat different error test procedure was used. The results are shown in Table 2 and speak for themselves.

Monkey neural data

We have decided to include the result on this dataset due to its interesting and

Rank	Classifier	Testing Mode	Test Error Rate %
1	**Nested Cavities (NC)**	**Cross validation**	**7.9**
2	CART-DB	Cross validation	10.0
3	**Nested Cavities (NC)**	**Train & Test**	**10.5**
4	**Enclosed Cavities (EC)**	**Cross validation**	**13.9**
5		**Train & Test**	**13.9**
6	CART	Cross validation	21.8
7	k-NN	Train & Test	44.0
8	RBF	Train & Test	46.5
9	Multi-layer perceptron	Train & Test	49.4
10	Single-layer perceptron	Train & Test	66.7

Table 2: Summary of classification results for the vowel dataset.

unusual features. Here there are two classes to be distinguished consisting of pulses measured on two separate neurons in a monkey's brain (poor thing!). The experiment was conducted at the Yale Medical School and we received the data from Prof. R. Coiffman's group which has been working on this classification problem. There are 600 samples with 32 variables. Remarkably, convergence was obtained with only one iteration and dimensionality selection required only 8 of the 32 parameters. The resulting ordering shows a striking separation. In the attached figure the first pair of variables x_1, x_2 originaly given is plotted showing no separation. In the adjoining plot the best pair x_{11}, x_{14}, as chosen by the classifier's ordering, shows remarkable separation. The discovery of this manually would require constructing and inspecting a scatterplot with 496 pairs ...! The result shows that the data consists of two "banana-like"[2] clusters in 8-D one (the complement in this case) enclosing the other (class for which the rule was found). Note that the classifier can actually describe highly complex regions. It can build and "carve" the cavity shown. It is no wonder that separation attempts in terms of hyperplanes or nearest-neighbour techniques can fail badly on such datasets. The rule gave an error of 3.92 % using train-and-test with 66 % of the data for training) and impressed the Yale group – not an easy feat!

Summary and Conclusions

- The **Nested Cavities (NC)** with the smaller number of steps and the larger number of variables gives, not surprisingly, constistently better results than the **Enclosed Cavities (EC)** version of the classifier.
- The larger the dataset the better the classification results.
- The classifier works best with continuous variables though it can handle well a small (i.e. no more than 20 % of the total) number of categorical variables.

The geometric formulation combined with the results on the representation of multidimensional objects in ||-coords gave a classifier with remarkably low computational complexity. This makes feasible the classification of truly large in size and number of variable datasets something we hope to test with suitable partners in the near future. The low complexity, enables the derivation of the rule in near real-time, and *then* apply it to incoming data, rendering the classifier **adaptive** to changing conditions. The rules provided are explicit, and "visualizable" and yield dimensionality selection which choses and orders the minimal set of variables needed to state the rule **without loss of information**. As it often happens such work raises new questions, on termination criteria, automatic approaches to overfiting, interpretation of the "geometry" of the dataset

[2] One observer suggested that this was due to the monkey's thinking of bananas during this fateful experiment ...!

as described by the rule and others. Automation with of Knowledged Discovery with parallel coordinates in finally in sight.

Internet Repositories

1. ftp.ics.uci.edu/pub/machine-learning-databases
2. ftp.ics.uci.edu/pub/machine-learning-databases/statlog
3. clyde.boltz.cs.cmu.edu/bench.html

References

[1] Deterding D. H. *Speaker Normalization for Automatic Speech Recognition*. Ph.D. Thesis, Cambridge University, 1989.

[2] Fayad U. M. Smyth P., Piatesky-Shapiro G. and Uthurusamy R. *Advances in Knowledge Discovery and Data Mining*. AAAI/MIT Press, 1996.

[3] Inselberg A. *Multidimensional Detective, in Proc. of IEEE Information Visualization '97, 100-107*. IEEE Comp. Soc., Los Alamitos, CA, 1997.

[4] Inselberg A. Don't panic ... do it in parallel. *J. of Comp. Stat.*, 14:53–77, 1999.

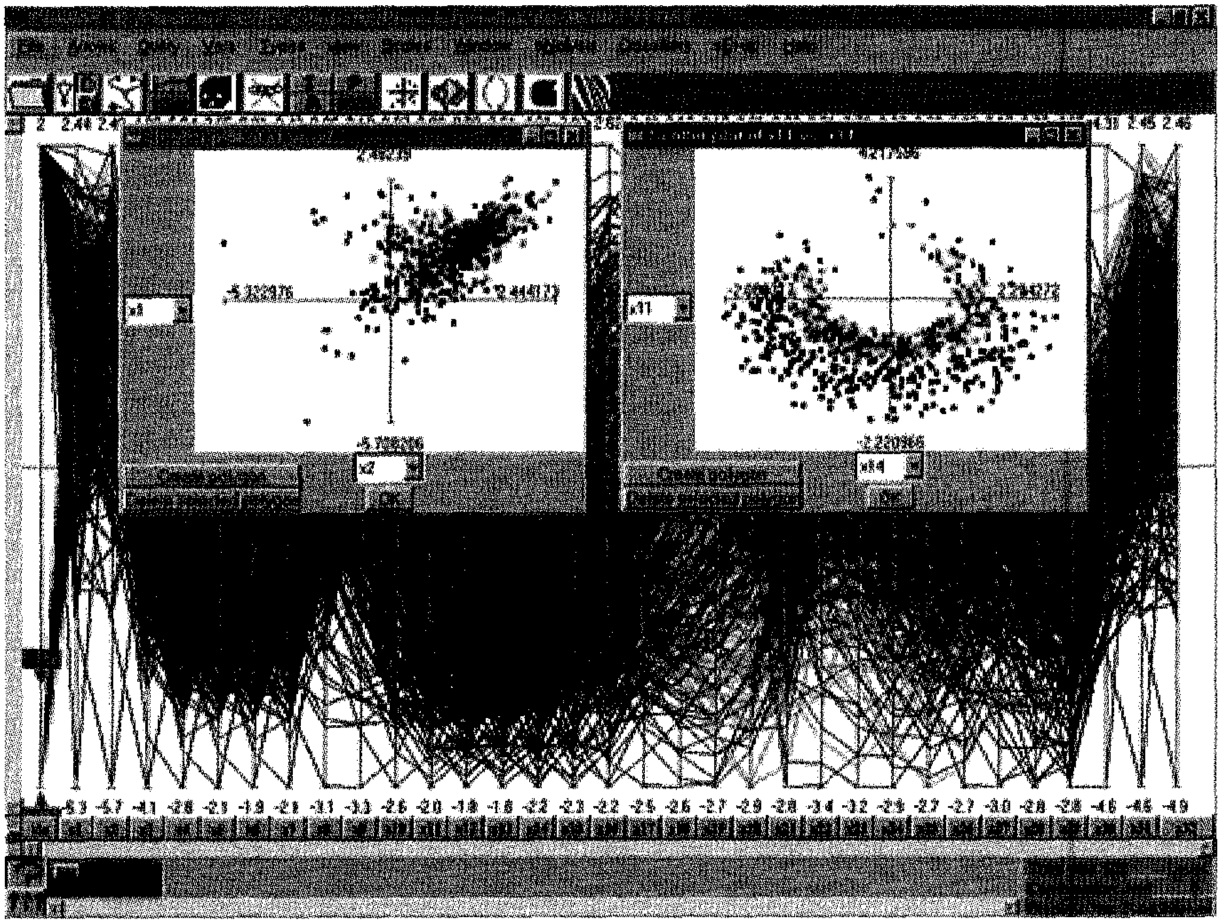
[5] Michie D. Spiegelhalter D.J. and Taylor C.C. *Machine Learning , Neural and Statistical Classification*. Ellis Horwood series in AI, 1994.

[6] Mitchell T.M. *Machine Learning*. McGraw-Hill, 1997.

[7] Quinlan J.R. *C4.5 : Programs for Machine Leaarning*. Morgan Kaufman, 1993.

[8] Robinson A. J. and Fallside F. *A Dynamic Connectionist Model of Phoneme Recognition*. Proc. of 1st European Neural Network Conf. (nEURO), 1988.

[9] Shang N. and Breiman L. *Distribution based trees are more accurate*. Proc. of 1st Inter. Conf. on Neural Info. Proc. (ICONIP96), 133, 1996.


-5.333976
2.448173
x2
-5.758266
Create polygon
Create selected polygon
OK
x31
x34
-2.220966
OK
-5.3 -5.7 -4.1 -2.8 -2.3 -1.9 -2.1 -3.1 -3.3 -2.6 -2.0 -1.9 -1.8 -2.2 -2.3 -2.2 -2.5 -2.6 -2.7 -2.9 -2.8 -3.4 -3.2 -2.9 -2.7 -2.7 -3.0 -2.8 -2.8 -4.6 -4.5 -4.9