Separability Index in Supervised Learning

Djamel A. Zighed, Stéphane Lallich, and Fabrice Muhlenbach

ERIC Laboratory – University of Lyon 2
5, av. Pierre Mendès-France, F-69676 BRON Cedex – FRANCE
{zighed,lallich,fmuhlenb}@univ-lyon2.fr

Abstract. We propose a new statistical approach for characterizing the class separability degree in R^p. This approach is based on a nonparametric statistic called "the Cut Edge Weight". We show in this paper the principle and the experimental applications of this statistic. First, we build a geometrical connected graph like the Relative Neighborhood Graph of Toussaint on all examples of the learning set. Second, we cut all edges between two examples of a different class. Third, we calculate the relative weight of these cut edges. If the relative weight of the cut edges is in the expected interval of a random distribution of the labels on all the neighborhood graph's vertices, then no neighborhood-based method will give a reliable prediction model. We will say then that the classes to predict are non-separable.

1 Introduction

Learning methods are very often requested in the data mining domain. The learning methods try to generate a prediction model φ from a learning sample Ω_l. Due to its construction method, the model is more or less reliable. This reliability is generally evaluated with *a posteriori* test sample Ω_t. The reliability depends on the learning sample, on the underlying statistical hypothesis, and on the implemented mathematical tools. Nevertheless, sometimes it does not exist any method that produce a reliable model, which can be explained by the following reasons:

- methods are not suitable to the problem we are trying to learn, so we have to find another method more adapted to the situation;
- the classes are not separable in the learning space. In this case, it is impossible to find a better learning method.

It will be very interesting to use mathematical tools that can characterize the class separability from a given learning sample. There already exist measures for learnability such as the VC-dimension provided by the statistical learning theory [20]. Nevertheless, VC-dimension is difficult to calculate in many cases. This problem has also been studied based on a statistical approach by Rao [16]. In the case of a normal distribution of the classes, Rao measures the learning ability degree through a test based on the population homogeneity. In a similar case, Kruskall and Wallis have defined a nonparametric test based on an equality

T. Elomaa et al. (Eds.): PKDD, LNAI 2431, pp. 475–487, 2002.

hypothesis of the scale parameters [1]. Recently, Sebban [18] and Zighed [23] have proposed a test based on the number of edges that connect examples of different classes in a geometrical neighborhood.

At first, they build a neighborhood structure by using some particular models like the Relative Neighborhood Graph of Toussaint [19]. After that, they calculate the number of edges that must be removed from the neighborhood graph to obtain clusters of homogeneous points in a given class. At last, they have established the law of the edge proportion that must be removed under the null hypothesis, denoted H_0, of a random distribution of the labels. With this law, they can say if classes are separable or not by calculating the p-value of the test –e.g., the probability of having a calculated value as important as the observed value under H_0.

In a more general view, we propose in this paper a theoretical framework and a nonparametric statistic that takes into consideration the weight of the removed edges. We exploit the works of the spatial autocorrelation, in particular the join-counts statistic, presented by Cliff and Ord [4] following the works of Moran [14], Krishna Iyer [9], Geary [7] and David [5]. Such process has been studied in the classification domain by Lebart [11] who used works based on the spatial contiguity, like the contiguity coefficient from Geary, to compare the local structures vs. the global structures in a k nearest neighbor graph.

To evaluate a learning method several points have to be distinguished. First, the quality of the results produced by the method have to be described, e.g., the determination coefficient R^2 in regression. Second, we have to test the hypothesis of the non-significance of the results. According to the number of instances, it could be known if the same value of R^2 is significant or not. Third, the robustness can be studied and the outliers can be searched. We propose a process that deals with all the previous points.

2 Class Separability, Clusters and Cut Edges

2.1 Notations

Machine learning methods intended to produce a function φ –like "decision rules" in the knowledge data discovery domain– that can predict the unknown belonging class $Y(\omega)$ of an instance ω extracted from the global population Ω, by knowing its representation $X(\omega)$.

In general, this representation $X(\omega)$ is provided by an expert who establishes *a priori* a set of attributes denoted: $X_1, X_2, ..., X_p$. Let these attributes take their values in R, $X : \omega \in \Omega \mapsto X(\omega) = (X_1(\omega), X_2(\omega), \ldots, X_p(\omega)) \in R^p$.

Within our context, all learning methods Φ must have recourse to a learning sample Ω_l and a test sample Ω_t. The former will be used for generating the prediction function φ, the latter will test the reliability of φ.

For all example $\omega \in (\Omega_l \cup \Omega_t)$, we suppose that its representation $X(\omega)$ and class $Y(\omega)$ are known. $Y : \Omega \mapsto \{y_1, \cdots, y_k\}$, with k the number of classes of Y.

The learning ability of a method is strongly associated to its class separability degree in $X(\Omega)$. We consider that the classes will be easier to separate if they fulfill the following conditions:

- the instances of the same class appear mostly gathered in the same subgroup in the representation space;
- the number of groups are so small, at least it reaches the number of the classes;
- the borders between the groups are not complex.

2.2 Neighborhood Graphs and Clusters

To express the proximity between examples in the representation space, we use the "neighborhood graph" notion [23]. These graphs are the Relative Neighborhood Graph (RNG), the Gabriel Graph, the Delaunay Triangulation and the Minimal Spanning Tree, that all provide planar and connected graph structures. We use here the RNG of Toussaint [19] defined below.

Definition: Let V be a set of points in a real space R^p (with p the number of attributes). The Relative Neighborhood Graph (RNG) of V is a graph with vertices set V, and the set of edges of the RNG of V are exactly those pairs (a, b) of points for which $d(a, b) \leq Max\left(d\left(a, c\right), d\left(b, c\right)\right) \forall c, c \neq a, b$, where $d(u, v)$ denotes the distance between two points u and v in R^p.

This definition means that the *lune* $L_{(u,v)}$ –constituted by the intersections of hypercircles centered on u and v with range the edge (u, v)– is empty. For example, on Fig. 1 (a), vertices 13 and 15 are connected because there is no vertex on the *lune* $L_{(13,15)}$.

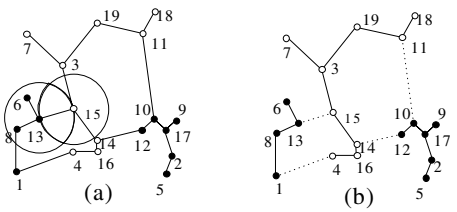

Fig. 1. RNG and clusters with two classes: the black and the white points

According to Zighed and Sebban [23] we introduce the concept of "cluster" to express that a set of close points have the same class. We call *cluster* a connected sub-graph of the neighborhood graph where all vertices belong to the same class. To build all clusters required for characterizing the structures of the scattered data points, we proceed in two steps:

1. we generate the geometrical neighborhood graph on the learning set;

2. we remove the edges connecting two vertices belonging to different classes, obtaining connected sub-graphs where all vertices belong to the same class.

The number of generated clusters gives a partial information on the class separability. If a number of clusters is low –at least the number of classes–, the classes are well separable and we can find a learning method capable of exhibit the model that underlies the particular group structure. For example on Fig. 1 (b), after cutting the four edges connecting vertices of different colors (in dotted line), we obtain three clusters for the two classes. But if this number tends to increase, closely to the number of clusters that we could have in a random situation, the classes could no longer be learned cause to the lack of a non random geometrical structure.

Actually, this number of clusters cannot ever characterize some little situations that seems intuitively different. For the same number of clusters, the situation can be very different depending on wether the clusters are easily isolated in the neighborhood graph or not. As soon as $p > 1$, rather than studying the number of clusters, we prefer to take an interest in the edges cut for building the clusters and we will calculate the relative weight of these edges in the edge set. In our example on Fig. 1 (b), we have cut four edges for isolating three clusters.

3 Cut Edge Weight Statistic

In a common point between supervised classification and spatial analysis, we consider a spatial contiguity graph which plays the role of the neighborhood graph [4]. The vertices of this graph are colored with k distinct colors. The color plays the role of the class Y. The matter is (1) to describe the link between the adjacency of two vertices and the fact they have the same color, and (2) to test the hypothesis of non significance. This would take us to test the hypothesis of no spatial autocorrelation between the values taken by a categorical variable over spatial units. In the case of a neighborhood graph, this would be the results for testing the hypothesis that the class Y cannot be learned from neighborhood-based methods.

3.1 Statistical Framework

Notations and Abbreviations

- Number of nodes in the graph: n
- Connection matrix: $V = (v_{ij}), i = 1, 2, ..., n; j = 1, 2, ..., n$; where $v_{ij} = 1$ if i and j are linked by an edge
- Weight matrix: $W = (w_{ij}), i = 1, 2, ..., n; j = 1, 2, ..., n$; where w_{ij} is the weight of edge (i, j). Let w_{i+} and w_{+j} be the sums of raw i and column j. We consider that W matrix is symmetrical. If we have to work with a non symmetrical matrix W', which is very interesting for neighborhood graphs, we will go back to the symmetrical case without loss of generality calculating: $w_{ij} = \frac{1}{2}\left(w'_{ij} + w'_{ji}\right)$.

– Number of edges: a
– Proportion of vertices corresponding to the class y_r: π_r, $r = 1, 2, ..., k$

According to Cliff and Ord [4], we adopt the simplified notations below:

Notations	Definition	Case : $W = V$
$\sum_2 w_{ij}$	$\sum_{i=1}^{n} \sum_{j=1, i\neq j}^{n} w_{ij}$	$2a$
S_0	$\sum_2 w_{ij}$	$2a$
S_1	$\frac{1}{2} \sum_2 (w_{ij} + w_{ji})^2$	$4a$
S_2	$\sum_{i=1}^{n} (w_{i+} + w_{+i})^2$	$4 \sum_{i=1}^{n} v_{i+}^2$

Definition of the Cut Edge Weight Statistic In order to take into consideration a possible weighting of the edges, we deal with the symmetrized weights matrix W which is reduced to the connection matrix V if all the weights are equal to 1.We consider both the symmetrical weights based upon the distances and non symmetrical weights based upon the ranks. In the case of distances, we choose $w_{ij} = (1 + d_{ij})^{-1}$, while in the case of ranks we choose $w_{ij} = \frac{1}{r_j}$, where r_j is the rank of the vertex j among the neighbors of the vertex i.

Edges linking two vertices of the same class (non cut edges) have to be distinguished from those linking two vertices of different classes (cut edges in order to obtain clusters).

Let us denote by I_r the sum of weights relative to edges linking two vertices of class r, and by $J_{r,s}$ the sum of weights relative to edges linking a vertex of class r and a vertex of class s. Statistics I and J are defined as it follows.

non cut edges	cut edges
$I = \sum_{r=1}^{k} I_r$	$J = \sum_{r=1}^{k-1} \sum_{s=r+1}^{k} J_{r,s}$

In so far as I and J are connected by the relation $I + J = \frac{1}{2} S_0$, we have only to study J statistic or its normalization $\frac{J}{I+J} = \frac{2J}{S_0}$. Both give the same result after standardization. We may observe that I generalizes the test of runs in 2 dimensions and k groups [13,21].

Random Framework Like Jain and Dubes [8], we consider binomial sampling in which null hypothesis is defined by:

H_0 : the vertices of the graph are labelled independently of each other, according to the same probability distribution (π_r) where π_r denotes the probability of the class $r, r = 1, 2, ..., k$.

We could consider hypergeometric sampling by adding into null hypothesis the constraint to have n_r vertices of the class $r, r = 1, 2, ...k$.

Rejecting null hypothesis means either the classes are non independently distributed or the probability distribution of the classes is not the same for the different vertices. In order to test the null hypothesis H_0 using statistic J (or I), we had first to study the distribution of these statistics under H_0.

3.2 Distribution of I and J under Null Hypothesis

To test H_0 with the statistic J, we will use two-sided test if we are surprised at once by abnormally small values of J (great separability of the classes) and by abnormally great values (deterministic structuration or pattern presence). Hypothesis H_0 is rejected when J produce an outstanding value taking into account its distribution under H_0. So, we have to establish the distribution of J under H_0 in order to calculate the p-value associated with the observed value of J as well as to calculate the critical value of J at the significance level α_0. This calculation can be done either by simulation or by normal approximation. In the last case, we have to calculate the mean and the variance of J under H_0.

Boolean Case The two classes defined by Y are noted 1 and 2. According to Moran [14], $U_i = 1$ if the class of the i^{th} vertex is 1 and $U_i = 0$ if the class is 2, $i = 1, 2, ..., n$. We denote π_1 the vertex proportion of class 1 and π_2 the vertex proportion of class 2. Thus:

$$J_{1,2} = \frac{1}{2}\sum_{2} w_{ij}\left(U_i - U_j\right)^2 = \frac{1}{2}\sum_{2} w_{ij} Z_{ij}$$

where U_i are independently distributed according to Bernoulli distribution of parameter π_1, noted $B(1, \pi_1)$. It must be noticed that the variables $Z_{ij} = \left(U_i - U_j\right)^2$ are distributed according to the distribution $B(1,\ 2\pi_1\pi_2)$, but are not independent. Actually, the covariances $Cov(Z_{ij}, Z_{kl})$ are null only if the four indices are different. Otherwise, when there is a common index, one can obtain:

$$Cov(Z_{ij}, Z_{il}) = \pi_1\pi_2(1 - 4\pi_1\pi_2)$$

The table below summarizes the different results related to the statistic $J_{1,2}$:

Variable	Mean	Variance
U_i	π_1	$\pi_1\pi_2$
$Z_{ij} = \left(U_i - U_j\right)^2$	$2\pi_1\pi_2$	$2\pi_1\pi_2\left(1 - 2\pi_1\pi_2\right)$
$J_{1,2}$	$S_0\pi_1\pi_2$	$S_1\pi_1^2\pi_2^2 + S_2\pi_1\pi_2\left(\frac{1}{4} - \pi_1\pi_2\right)$
$J_{1,2}$ si $w_{ij} = v_{ij}$	$2a\pi_1\pi_2$	$4a\pi_1^2\pi_2^2 + \pi_1\pi_2\left(1 - 4\pi_1\pi_2\right)\sum_{i=1}^{n} v_{i+}^2$

The p-value of $J_{1,2}$ is calculated from standard normal distribution after centering and reducing its observed value. The critical values for $J_{1,2}$ at the significance level α_0 are:

$$J_{1,2;\alpha_{0/2}} = S_0\pi_1\pi_2 - u_{1-\alpha_{0/2}}\sqrt{S_1\pi_1^2\pi_2^2 + S_2\pi_1\pi_2\left(\frac{1}{4} - \pi_1\pi_2\right)}$$

$$J_{1,2;1-\alpha_{0/2}} = S_0\pi_1\pi_2 + u_{1-\alpha_{0/2}}\sqrt{S_1\pi_1^2\pi_2^2 + S_2\pi_1\pi_2\left(\frac{1}{4} - \pi_1\pi_2\right)}$$

By simulation, the most convenient is to calculate the p-value associated with the observed value of $J_{1,2}$. To simulate a realization of $J_{1,2}$, one only has to simulate a realization of $B(1, \pi_1)$ for each example, which requires n random numbers between 0 and 1, and then to apply the formula which defines $J_{1,2}$. After having repeated N times the operation, one calculates the p-value associated with the observed value of $J_{1,2}$ by calculating the proportion of simulated values of $J_{1,2}$ which are less or equal to the observed value of $J_{1,2}$.

Multiclass Case To extend these results to the multiclass case, according to Cliff and Ord [4], we reason with I and J statistics already defined. These statistics are:

$$\boxed{I = \sum_{r=1}^{k} I_r = \frac{1}{2} \sum_2 w_{ij} T_{ij}} \quad \boxed{J = \sum_{r=1}^{k-1} \sum_{s=r+1}^{k} J_{r,s} = \frac{1}{2} \sum_2 w_{ij} Z_{ij}}$$

where T_{ij} and Z_{ij} are random boolean variables which indicate if the vertices i and j have the same class (T_{ij}) or not (Z_{ij}).

From previous results, we easily obtain the mean of I and J:

Test statistic	Mean
$I = \sum_{r=1}^{k} I_r$	$\frac{1}{2} S_0 \sum_{r=1}^{k} \pi_r^2$
$J = \sum_{r=1}^{k-1} \sum_{s=r+1}^{k} J_{r,s}$	$S_0 \sum_{r=1}^{k-1} \sum_{s=r+1}^{k} \pi_r \pi_s$

Because I and J are connected by the relation $I + J = \frac{1}{2} S_0$, these two variables have the same variance, denoted $\sigma^2 = Var(I) = Var(J)$. The calculation of σ^2 is complicated due to the necessity of taking the covariances into consideration. In accordance with Cliff and Ord [4], we obtain the following results for binomial sampling:

$$4\sigma^2 = S_2 \sum_{r=1}^{k-1} \sum_{s=r+1}^{k} \pi_r \pi_s + (2S_1 - 5S_2) \sum_{r=1}^{k-2} \sum_{s=r+1}^{k-1} \sum_{t=s+1}^{k} \pi_r \pi_s \pi_t$$
$$+4 (S_1 - S_2) \left[\sum_{r=1}^{k-1} \sum_{s=r+1}^{k} \pi_r^2 \pi_s^2 - 2 \sum_{r=1}^{k-3} \sum_{s=r+1}^{k-2} \sum_{t=s+1}^{k-1} \sum_{u=t+1}^{k} \pi_r \pi_s \pi_t \pi_u \right]$$

3.3 Complexity of the Test

Differents steps are into consideration: computing the matrix distance is in $O(p \times n^2)$, with n the number of examples and p the attributes, and building the neighborhood graph in R^p is in $O(n^3)$. Because the number of attributes p is very small compared to the number of instances n, the test is in $O(n^3)$.

We point out that all the complete database in not needed for the test. A sample, particulary a stratified sample, can be enough to reveal a good idea of the class separability of the database.

4 From Numerical Attributes to Categorical Attributes

We have introduced the test of weighted cut edges for Y is a categorical variable and the attributes $X_1, X_2, ..., X_p$ are numerical. One notice that in order to apply such a test in a supervised learning, we only need to build the neighborhood

graph which summarizes the information brought by the attributes. To the extent that the building of this neighborhood graph only requires the dissimilarity matrix between examples, we may consider a double enlargement of the weighted cut edge test.

The first enlargement corresponds to the situation of categorical attributes $X_j, j = 1, 2, ..., p$, which often exists in the real world. In such a case, it is enough to construct a dissimilarity matrix from the data.

We have to use a dissimilarity measure suited to the nature of attributes (cf. Chandon and Pinson [3], Esposito et al. [6]).

In the case of boolean data, there is a set of similarity indices between examples relying on the number of matching "1" (noted by a) or "0" (d) and the number of mismatching "1-0" (b) or "0-1" (c). A general formula for similarity ($s_{\theta_1\theta_2}$) and dissimilarity ($d_{\theta_1\theta_2}$) indices taking their values between 0 and 1, is:

$$s_{\theta_1\theta_2} = \frac{a + \theta_1 d}{a + \theta_1 d + \theta_2(b + c)} = 1 - d_{\theta_1\theta_2}$$

Most known indices are mentioned in the table above.

Table 1. Main similarity indices

θ_1	θ_2	Name
1	1	Sokal and Michener, 1958
1	2	Rogers and Tanimoto, 1960
1	0.5	not named
0	1	Jaccard, 1900
0	2	Sokal and Sneath, 1963
0	0.5	Czekanowski, 1913; Dice, 1945

In the case of categorical data, there are two main methods:

- either to generalize the previously quoted indices when it is possible. For example, Sokal and Michener index is the proportion of matching categorical attributes. It is possible to weight the attributes according to their number of categorical components.
- or to rewrite each categorical attribute as a set of boolean attributes in order to use indices for boolean data. In this case, all the examples have the same number of "1", namely p. Then, according to Lerman (1970), all the indices mentioned in Table 1 lead to the same ordering on the set of example's pairs. Applying Minkowski distance of parameter 1 or 2 to such a matrix is equivalent to the generalization of Sokal and Michener index.

Lastly, when variables are of different types, one can use a linear weighted combination of dissimilarity measures adapted to each type of variable or reduce the data to the same type [6].

The second enlargement deals with the situation where only a dissimilarity matrix D is known and not the original data X. This situation arises for instance in the case of input-output tables (e.g., Leontiev Input-output table) or when the collected information is directly dissimilarity matrix (e.g., in marketing or psychology trials).

5 Experiments

5.1 Cut Weighted Edge Approach for Numerical Attributes

Values of the Cut Weighted Edge Test The weighted edge test has been experimentally studied on 13 benchmarks from the UCI Machine Learning Repository [2]. These databases have been chosen for having only numerical attributes and a symbolic class. For each base, we build a relative neighborhood graph [19] on the n instances of the learning set. In Table 1, the results show the number of instances n, the number of attributes p and the number of classes k. We present also information characterizing the geometrical graph: number of obtained edges for constructing the graph (*edges*) and the number of cluster obtained after cutting the edges linking two vertices of different classes (*clusters*).

Table 2. Cut weighted edge test values on 13 benchmarks

General information							without weighting			weighting: distance			weighting: rank		
Domain name	n	p	k	clust.	edges	error r.	J / (I + J)	J^s	p-value	J / (I + J)	J^s	p-value	J / (I + J)	J^s	p-value
Wine recognition	178	13	3	9	281	0.0389	0.093	-19.32	0	0.054	-19.40	0	0.074	-19.27	0
Breast Cancer	683	9	2	10	7562	0.0409	0.008	-25.29	0	0.003	-24.38	0	0.014	-25.02	0
Iris (Bezdek)	150	4	3	6	189	0.0533	0.090	-16.82	0	0.077	-17.01	0	0.078	-16.78	0
Iris plants	150	4	3	6	196	0.0600	0.087	-17.22	0	0.074	-17.41	0	0.076	-17.14	0
Musk "Clean1"	476	166	2	14	810	0.0650	0.167	-17.53	0	0.115	-7.69	2E-14	0.143	-18.10	0
Image seg.	210	19	7	27	268	0.1238	0.224	-29.63	0	0.141	-29.31	0	0.201	-29.88	0
Ionosphere	351	34	2	43	402	0.1397	0.137	-11.34	0	0.046	-11.07	0	0.136	-11.33	0
Waveform	1000	21	3	49	2443	0.1860	0.255	-42.75	0	0.248	-42.55	0	0.248	-42.55	0
Pima Indians	768	8	2	82	1416	0.2877	0.310	-8.74	2E-18	0.282	-9.86	0	0.305	-8.93	4E-19
Glass Ident.	214	9	6	52	275	0.3169	0.356	-12.63	0	0.315	-12.90	0	0.342	-12.93	0
Haberman	306	3	2	47	517	0.3263	0.331	-1.92	0.0544	0.321	-2.20	0.028	0.331	-1.90	0.058
Bupa	345	6	2	50	581	0.3632	0.401	-3.89	0.0001	0.385	-4.33	1E-05	0.394	-4.08	5E-05
Yeast	1484	8	10	401	2805	0.4549	0.524	-27.03	0	0.512	-27.18	0	0.509	-28.06	0

On Table 2, for each base, we present the relative cut edge weight $\frac{J}{I+J}$, the standardized cut weighted edge test J^s with its p-value in three cases: when the test is done without weighting, when the edges are weighted by the inverse of the distance between the vertices, and when the edges are weighted by the inverse of the number of the rank of a vertex to the others of the graph. For each base and weighting method, the p-values are extremely low, this shows that the null hypothesis of a random distribution of the labels on the vertices of a neighborhood graph is very strong.

For information, the empirical evaluation of the CPU time needed for the test (distance matrix computation, graph construction, edges cut, test statistic

calculation) is between a little less than 1 second for *Iris* (150 instances) and 200 seconds for *Yeast* (about 1,500 instances) on a 450 MHz PC. We present only the results obtained with a RNG graph of Toussaint (the results with a Gabriel Graph or a Minimal Spanning Tree are very close to them).

Weight of the Cut Edges and Error Rate in Machine Learning The 13 benchmarks have been tested on the following different machine learning methods: instance-based learning method (the nearest neighborhood: 1-NN [12]), decision tree (C4.5 [15]), induction graph (Sipina [22]), artificial neural networks (Perceptron [17], Multi-Layer Perceptron with 10 neurons on one hidden layer [12]) and the Naive Bayes [12]. On Table 3 we present the error rates obtained by these methods on a 10 cross validation with the benchmarks and the statistical values previously calculated (without weighting). The rate errors for the different learning methods, and particulary the mean of these methods, are well correlated with the relative cut edge weight ($J/(I+J)$). We can see on Fig. 2 the linear relation between the relative cut edge weight and the mean of the error rate for the 13 benchmarks.

Table 3. Error rates and statistical values of the 13 benchmarks

	General information					Statistical value			Error rate						
Domain name	n	p	k	clust.	edges	J/(I+J)	J^s	p-value	1-NN	C4.5	Sipina	Perc.	MLP	N. Bayes	Mean
Breast Cancer	683	9	2	10	7562	0.008	-25.29	0	0.041	0.059	0.050	0.032	0.032	0.026	0.040
BUPA liver	345	6	2	50	581	0.401	-3.89	0.0001	0.363	0.369	0.347	0.305	0.322	0.380	0.348
Glass Ident.	214	9	6	52	275	0.356	-12.63	0	0.317	0.289	0.304	0.350	0.448	0.401	0.352
Haberman	306	3	2	47	517	0.331	-1.92	0.0544	0.326	0.310	0.294	0.241	0.275	0.284	0.288
Image seg.	210	19	7	27	268	0.224	-29.63	0	0.124	0.124	0.152	0.119	0.114	0.605	0.206
Ionosphere	351	34	2	43	402	0.137	-11.34	0	0.140	0.074	0.114	0.128	0.131	0.160	0.124
Iris (Bezdek)	150	4	3	6	189	0.090	-16.82	0	0.053	0.060	0.067	0.060	0.053	0.087	0.063
Iris plants	150	4	3	6	196	0.087	-17.22	0	0.060	0.033	0.053	0.067	0.040	0.080	0.056
Musk "Clean1"	476	166	2	14	810	0.167	-17.53	0	0.065	0.162	0.232	0.187	0.113	0.227	0.164
Pima Indians	768	8	2	82	1416	0.310	-8.74	2.4E-18	0.288	0.283	0.270	0.231	0.266	0.259	0.266
Waveform	1000	21	3	49	2443	0.255	-42.75	0	0.186	0.260	0.251	0.173	0.169	0.243	0.214
Wine recognition	178	13	3	9	281	0.093	-19.32	0	0.039	0.062	0.073	0.011	0.017	0.186	0.065
Yeast	1484	8	10	401	2805	0.524	-27.03	0	0.455	0.445	0.437	0.447	0.446	0.435	0.444
								Mean	0.189	0.195	0.203	0.181	0.187	0.259	0.202
								R^2 (J/(I+J) ; error rate)	0.933	0.934	0.937	0.912	0.877	0.528	0.979
								R^2 (J^s ; error rate)	0.076	0.020	0.019	0.036	0.063	0.005	0.026

5.2 Complementary Experiments

Cut Weighted Test and Categorical Attributes To show how to deal with categorical attributes, we have applied the cut weighted edge test on the benchmark *Flag* of the UCI Repository [2] that contains such predictors (Table 4). The categorical attributes have been rewritten as a set of boolean attributes and the neighborhood graph is build with all standardized attributes. The test indicates that this base is separable, related to the mean error rate of 0.36 for 6 classes to learn.

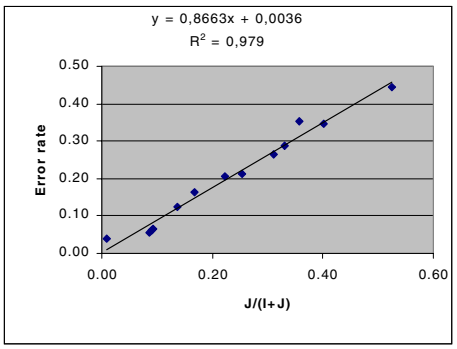

Fig. 2. Relative cut edge weight and mean of the error rates

Size Effect of the Database We point out the fact that J^s, the standardized cut weighted edge statistic, and then the p-value depend strongly of the size of the learning set. The same observed deviation in the null hypothesis is more significant, because of the learning set size. This fact is illustrated by the results of experiments conducted on the benchmark waves, for different size of learning set (n=20, 50, 100, 1000). The results of the tests are shown on Table 4. The error rates are decreasing but we do not present their values in the different learning methods because of the great variability due to the small size of the learning set. The p-value is not significant for n=20, and it is more and more significant when n increases. Concurrently, we notice that $\frac{J}{I+J}$ decreases as well as the error rate.

Table 4. Error rates and statistical values of the other databases

Domain name	n	p	k	clust.	edges	J / (I + J)	J s	p-value	1-NN	C4.5	Sipina	Perc.	MLP	N. Bayes	Mean
Flag	194	67	6	46	327	0.489	-13.91	0	0.366	0.346	0.371	0.310	0.428	0.340	0.360
Waves-20	20	21	3	6	25	0.400	-0.44	0.6635							
Waves-50	50	21	3	11	72	0.375	-4.05	5.0E-05							
Waves-100	100	21	3	12	156	0.301	-8.44	3.3E-17							
Waves-1000	1000	21	3	49	2443	0.255	-42.75	0							

6 Conclusion

In this paper that proceeds the research of Zighed and Sebban [23], our results outcome a strict framework that permits to take into consideration the weight of the edges for numerical or categorical attributes. Furthermore we can use this framework to detect outliers and improve classification [10].

The construction of the test is based on the existence of a neighborhood graph. To build this graph, the dissimilarity matrix is only needed. This char-

acteristic gives to our approach a very general dimension to estimate the class separability, however the instance representation may be known or not.

Our perspectives are to describe the procedures of implementing and identifying the application fields, in order to make tests on real applications.

References

1. S. Aivazian, I. Enukov, and L. Mechalkine. *Eléments de modélisation et traitement primaire des données*. MIR, Moscou, 1986. 476
2. C. L. Blake and C. J. Merz. UCI repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science [http://www.ics.uci.edu/∼mlearn/MLRepository.html], 1998. 483, 484
3. J. L. Chandon and S. Pinson. *Analyse Typologique, Théories et Applications*. Masson, 1981. 482
4. A. D. Cliff and J. K. Ord. *Spatial processes, models and applications*. Pion Limited, London, 1986. 476, 478, 479, 481
5. F. N. David. Measurement of diversity. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability*, pages 109–136, Berkeley, USA, 1971. 476
6. F. Esposito, D. Malerba, V. Tamma, and H. H. Bock. Similarity and dissimilarity measures: classical resemblance measures. In H. H. Bock and E. Diday, editors, *Analysis of Symbolic data*, pages 139–152. Springer-Verlag, 2000. 482
7. R. C. Geary. The contiguity ratio and statistical mapping. *The Incorporated Statistician*, 5:115–145, 1954. 476
8. A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall, 1988. 479
9. P. V. A. Krishna Iyer. The first and second moments of some probability distribution arising from points on a lattice, and their applications. In *Biometrika*, number 36, pages 135–141, 1949. 476
10. S. Lallich, F. Muhlenbach, and D. A. Zighed. Improving classification by removing or relabeling mislabeled instances. In *Proceedings of the XIIIth Int. Symposium on Methodologies for Intelligent Systems (ISMIS)*, 2002. To appear in LNAI. 485
11. L. Lebart. Data anlysis. In W. Gaul, O. Opitz, and M. Schader, editors, *Contiguïty analysis and classification*, pages 233–244, Berlin, 2000. Springer. 476
12. T. Mitchell. *Machine Learning*. McGraw Hill, 1997. 484
13. A. Mood. The distribution theory of runs. *Ann. of Math. Statist.*, 11:367–392, 1940. 479
14. P. A. P. Moran. The interpretation of statistical maps. In *Journal of the Royal Statistical Society*, serie B, pages 246–251, 1948. 476, 480
15. J. R. Quinlan. *C4.5: Program for Machine Learning*. Morgan Kaufmann, San Mateo, Ca, 1993. 484
16. C. R. Rao. *Linear statistical inference and its applications*. Wiley, New-York, 1972. 475
17. F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–408, 1958. 484
18. M. Sebban. *Modèles théoriques en reconnaissance des formes et architecture hybride pour machine perceptive*. PhD thesis, Université Lyon 2, 1996. 476
19. G. Toussaint. The relative neighborhood graph of a finite planar set. *Pattern recognition*, 12:261–268, 1980. 476, 477, 483

20. V. Vapnik. *Statistical Learning Theory*. John Wiley, NY, 1998. 475
21. A. Wald and J. Wolfowitz. On a test wether two samples are from the same population. *Ann. of Math. Statist.*, 11:147–162, 1940. 479
22. D. A. Zighed, J. P. Auray, and G. Duru. *SIPINA : Méthode et logiciel*. Lacassagne, 1992. 484
23. D. A. Zighed and M. Sebban. Sélection et validation statistique de variables et de prototypes. In M. Sebban and G. Venturini, editors, *Apprentissage automatique*. Hermès Science, 1999. 476, 477, 485