Complexity of Data Subsets Generated by the Random Subspace Method: An Experimental Investigation

Ludmila I. Kuncheva[1], Fabio Roli[2], Gian Luca Marcialis[2], and Catherine A. Shipp[1]

[1]School of Informatics, University of Wales, Bangor
Gwynedd, LL57 1UT, United Kingdom, {c.a.shipp,l.i.kuncheva}@bangor.ac.uk
[2]Dept. of Electrical and Electronic Eng., University of Cagliari
Piazza d'Armi, I-09123 Cagliari, Italy, {marcialis,roli}@diee.unica.it

Abstract

Recently, Ho described three measures of complexity of classification tasks and related them to the comparative advantages of two methods for creating multiple classifiers: the Bootstrap method and the Random Subspace method [3]. Here we report the results from an experimental investigation on the complexity of data subsets generated by the Random Subspace method. Our motivation is based on the assumption that multiple classifier systems achieve best results when the individual classifiers are of similar accuracy. Similar individual accuracies will be obtained on data sets of similar complexity. Hence, the main aim of this study is to analyse the *variability* of the complexity among the generated subsets. Four measures of complexity have been used, three from [3]: the minimal spanning tree (MST), the adherence subsets measure (ADH), the maximal feature efficiency (MFE); and a cluster label consistency measure (CLC) proposed in [7]. Continuing the study from [7], we used again the UCI "wine" data set (3 classes, 13 features) with 7 cases of class split: 1v2v3, 1v(2&3), 2v(1&3), and 3v(1&2), and the 3 pairwise splits. Our results relate the variability in data complexity to the number of features used, the presence of redundant features, and variability when using bootstrapping and data splitting to generate data sets for multiple classifier systems.

1 Introduction

Recently, Ho described three measures of complexity of classification tasks and related them to the comparative advantages of two methods for creating multiple classifiers, namely, the Bootstrap method and the Random Subspace method [3]. Here we report the results from a pilot experimental investigation on the complexity of data subsets generated by the Random Subspace method. The main aim was to analyse the *variability* of the complexity among the generated subsets. The rationale behind this objective is the assumption that multiple classifier systems achieve best results when the individual classifiers are *of similar accuracy.* The intuition for this statement is that if the individual classifiers are very different in accuracy (they must be *diverse* otherwise for best performance!), then there will be

a. at least one classifier which is much better than the rest of the team, and thus using the whole team will hardly improve on the best individual, or

b. at least one classifier which is much worse than the rest of the team, and using it in a combination will only degrade the overall performance.

In other words, it is reasonable to expect to gain from using a team of classifiers when the classifiers are of approximately the same accuracy even if this accuracy is not too high [4].

Another intuitive assumption is that data complexity is straightforwardly related to classification accuracy. Therefore, if the generating method produces subsets of similar complexity, we can expect that classifiers of similar accuracies can be built upon them. However, this does not mean that the classifiers will possess the necessary *diversity* to form a good team.

Note that the individual accuracy and team diversity are different concepts. The members of the team might have the same accuracy and be identical or be as as diverse as the accuracy allows for. For example, let D_1 and D_2 be classifiers of equal accuracies, run on 100 objects. Assume that each classifier recognizes 98 of the 100 objects. The classifiers might be the identical (failing on the same 2 objects), "semi-diverse" (failing simultaneously on 1 object and separately on 1 object each) or diverse (each one failing on a different couple of objects). The individual accuracy is clearly related to the complexity of the problem but diversity is not. Depending on how diversity is defined, it may be bounded from above, and the bound will depend on the magnitude of the accuracy (c.f. [5]). On

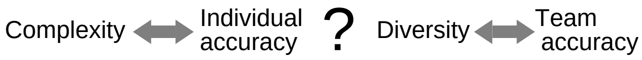

Figure 1: Illustration of a relationship

the other hand, the accuracy *of the team* is related to the diversity among the team members. Thus, we cannot expect a clear-cut relationship between the accuracy of the team and the complexity of the data sets on which the individual classifiers are designed (The conceptual relationship between the four notions is illustrated in Figure 1). Therefore we confine the study to finding out about the variability of complexity of the data sets and do not attempt to relate this variability to the team accuracy. An analysis on this matter is out of the scope of this paper.

2 Methods for generating data sets in multiple classifier systems

Unfortunately, independent training of classifiers do not guarantee independence (in statistical sense) of their outputs [6]. One approach to enhancing diversity of the individual classifiers is to train them on different subsets of the available labeled data set. Let $Z = \{\mathbf{z}_1, \ldots, \mathbf{z}_N\}$ be a labeled data set, $\mathbf{z}_j \in \Re^n$, $j = 1, \ldots, N$ with N elements. Three methods for obtaining different data subsets can be distinguished

1. **Random Subspace**. We can base the individual classifiers on different subsets of features, i.e., on different subspaces of the feature space $\Re^n$. Ho [2] shows that random sampling (without repetition) to get a set of $d < n$ features from the integers from 1 to n, is a viable line for building multiple classifier systems.

2. **Bootstrap sampling**. We sample from Z with repetition assuming uniform probability across the elements of Z. Typically the cardinality of the bootstrap sample is chosen to be N. Bootstrap sampling underlies the boosting method for designing classifier ensembles [1].

3. **Data splitting**. The individual classifiers can be built on disjoint subsets of Z, as in crossvalidation estimation of classification accuracy.

We applied four measures of complexity: the minimal spanning tree (MST), the adherence subsethood (ADH) based on the ϵ-neighborhood measure, the maximum feature efficiency (MFE), all three from [3], and a measure which we call the *Cluster Label Consistency (CLC)*, introduced in our previous study [7]. In this study we bring in some results from [7] and continue with an additional study on the variability in complexity of the data sets generated by the Random Subspace method.

3 Measures of complexity

3.1 Minimal Spanning Tree

Given the data set Z and a metric on $\Re^n$, a minimal spanning tree can be constructed which connects all the sample points regardless of their class labels. Here we use the Euclidean distance as the metric throughout this study. Clearly some edges of the MST will connect points from different classes and the count of such edges gives us a measure of the length of the boundary between the classes. Since there are $N-1$ edges for N sample points, the count can be expressed as a percentage of N [3], leading to a complexity measure

$$MSTcomplexity = \frac{N_e}{N}, \tag{1}$$

where N_e is the number of edges in the minimal spanning tree connecting different classes.

3.2 Adherence Subsets

This method proposed by Ho [3] considers the clustering properties of the data. It is based on a reflexive and symmetric (tolerance) binary relation $\mathcal{R}$ between two points $x,\ y$ in a set F. $\mathcal{R}$ is defined by $x\mathcal{R}y \Leftrightarrow d(x,y) < \epsilon$, where $d(x,y)$ is a given metric and ϵ is a given non-zero constant. We define $\Gamma(x) = \{y \in F | y\mathcal{R}x\}$ to be the ϵ-neighbourhood of x. An adherence mapping, ad from the power set $\mathcal{P}(F)$ to $\mathcal{P}(F)$ is such that:

$$\begin{cases} ad(\phi) & = & \phi \\ ad(x) & = & \Gamma(x) \\ ad(A) & = & \bigcup_{x \in A} ad(x) \quad \forall A \subseteq F. \end{cases}$$

The largest possible adherence subsets can be grown for each point by successively expanding the adherence subset at each stage whilst ensuring that all newly included points come from the same class. For example, $ad^0(\{x\}) = \{x\}$, $ad^1(\{x\}) = ad(\{x\})$, $ad^2(\{x\}) = ad(ad(\{x\}))\ldots$, gives us progressively higher order adherence subsets. For each point only the highest order subset is retained such that all elements are from the same class. This procedure defines a partition of the data set where each cluster contains data points with the same class label. The number of such clusters is an indication of the complexity of the problem. If the classes are compact and far from each other, then each class will ideally form a single separate adherence subset. When the classes are overlapping, multiple clusters are likely to appear.

The calculation works by taking a labelled data set Z of size N and for each point growing the largest possible adherence subset such that all elements of the subset are from the same class. The complexity is then given by:

$$ADHcomplexity = \frac{N_s}{N} \tag{2}$$

where N_s is the number of different adherence subsets..

In Ho's paper [3] the choice of ϵ was $\epsilon = 0.55\delta$ where δ was the minimal distance between two points of different classes. In a preliminary experiment we studied the effect of ϵ on the complexity value and found that the relationship between ϵ and $ADHcomplexity$ is not monotonic. Indeed, if ϵ is too small, then each point will be a cluster on its own, and $ADHcomplexity = 1$. On the other hand, if ϵ is too large, then the ϵ-neighbourhood of $\mathbf{x}$ will contain point(s) from a different class. Again, $\mathbf{x}$ will be marked as a cluster on its own, leading to $ADHcomplexity = 1$. Since there seems to be no clear reason for choosing a particular ϵ, we picked $\epsilon = min + 0.1 * (max - min)$, where min and max were the minimum and maximum distances in the data set regardless of class labels.

3.3 Maximum Feature Efficiency

This method is suitable for 2 classes only. The complexity on each feature is assessed separately. All points are projected on that feature axis and the overlap interval is found. The MFE complexity for the i-th feature is

$$MFEcomplexity_i = \frac{N_i}{N}, \tag{3}$$

where N_i is the number of points within the overlap interval. The final complexity value is defined as

$$MFEcomplexity = \min_i MFEcomplexity_i. \tag{4}$$

3.4 Cluster Label Consistency

This measure estimates how well the classes match the possible clusters in data. First c clusters are obtained on the whole data set regardless of the class labels and then the labels are used to count the number from each class within each cluster. "Pure" clusters will give low complexity values whereas "contaminated" clusters will give high complexity values. The complexity measure is

$$CLCcomplexity = 1 - \frac{1}{c}\sum_{i=1}^{c} C_i, \tag{5}$$

where C_i is the *cluster label consistency* of cluster i, found as the fraction of the maximal number of points of the same class label in the cluster. In case of a perfect match, i.e., when each class is a cluster on its own, the complexity is 0.

Consider as an example a data set distributed according to a mixture of 5 Gaussians in $\Re^2$ with centers $(0,0),(2,3),(0,4),(3,1)$ and $(2,4)$, respectively, and variance 0.4 along each axis. The left plot in Figure 2 shows the clusters and their centroids. A circular decision boundary is applied on the data set centered at $(1,2)$ with radius 2. All points inside the circle are labeled in class ω_1, and the points outside the circle are labeled in ω_2, as illustrated on the right plot in Figure 2. Thus, the Bayes error for this data model is zero. The peculiar feature about this data set is that the cluster structure of the data is not representative for the true class structure.

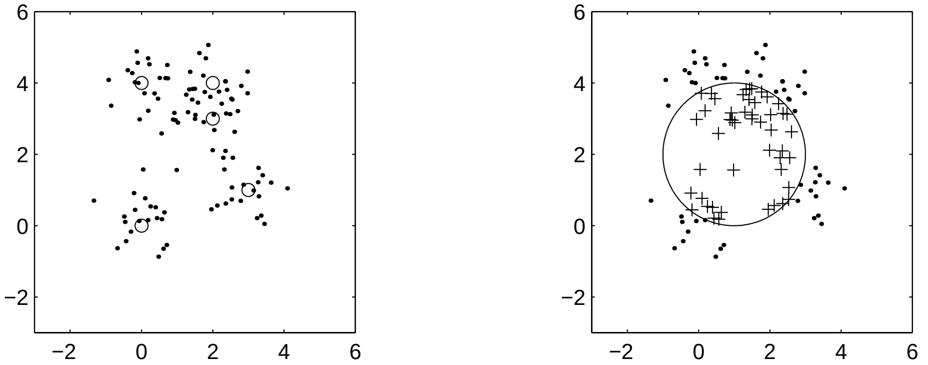

Figure 2: A 5-clusters example of a problem where perfect separation is possible but the complexity is high because the cluster structure of the data is not representative for the class label structure.

The quadratic discriminant classifier gave a 17 % training error on this data set. The following complexity values have been obtained:

$MST_{complexity}$	0.1200	$CLC_{complexity}$ $(c=2)$	0.4782
$ADH_{complexity}$	1.000	$CLC_{complexity}$ $(c=3)$	0.4789
$MFE_{complexity}$	0.6800	$CLC_{complexity}$ $(c=4)$	0.4408
		$CLC_{complexity}$ $(c=5)$	0.4632

The purpose of showing this example was to highlight two seemingly discouraging observations. First, clearly the *achievable* individual accuracy (100 % in this case) is not necessarily related with the measures of complexity. Second, the results give us an early indication of the severe disagreement between the measures of complexity despite the fact that they are all meant to measure the same characteristic of the data set. However, we note that: (1) in real problems, the class-cluster relationship may be less deceiving than in this example, and (2), the difference in the values of the complexity measures shows that the notion "complexity" needs a stricter definition beyond the common intuition.

The limits of the 4 measures of complexity for fixed c (classes) and N (elements) are shown in Table 1.

Table 1: Limits of the 4 measures of complexity for fixed c (classes) and N (elements).

Measure	MST	ADH	MFE	CLC
Lower limit	$\frac{c-1}{N}$	$\frac{c}{N}$	0	0
Upper limit	$\frac{N-1}{N}$	1	1	$1-\frac{1}{c^2}$

4 Experiments

4.1 Data

We used the "wine" data set from the UCI Repository of Machine Learning Database[1]. It contains 178 cases labeled in 3 classes, with 13 continuous-valued features and no missing values. From this data set we derived the following 7 problems

[1] Found at [http://www.ics.uci.edu/ mlearn/MLRepository.html]

Case A:	1 v 2 v 3.
Case B:	1 v (2 and 3).
Case C:	2 v (1 and 3).
Case D:	3 v (1 and 2).
Case E:	1 v 2.
Case F:	1 v 3.
Case G:	2 v 3.

Applying the Random Subspace method, we formed 50 subsets by randomly choosing 5 of the $n = 13$ features. The four complexity measures were calculated for each data set.

4.2 Results

First we calculated the complexity for the whole of the data set for the seven cases (Table 2). The results show that the $ADHcomplexity$ considers each problem to be of the same high degree of complexity, whereas the other three measures give different values. The four measures do not agree on a single case being the most complex or the easiest.

Table 2: Complexity calculated by MST, ADH, MFE and CLC (in %) for the data set as a whole

Case	MST	ADH	MFE	CLC
A	25	100	N/A	28
B	8	100	24	9
C	21	100	42	26
D	21	100	28	21
E	8	100	21	8
F	9	100	0	11
G	27	100	25	34

Table 3 shows the means and the standard deviations the 4 measures and the 7 cases. As in the example at the end of the previous section, the measures give very different values. Knowing that the three of them (except CLC) span approximately the same intervals ($0.01 \leq MSTcomplexity \leq 0.99$,

Table 3: Complexity calculated by MST, ADH, MFE and CLC (in %) with the Random Subspace method for the 7 cases

Case	MST		ADH		MFE		CLC	
	mean	std	mean	std	mean	std	mean	std
A	21	7	100	0	–	–	35	11
B	12	5	99	4	44	14	23	11
C	18	5	100	3	57	15	30	6
D	13	8	95	17	34	12	24	4
E	13	5	99	7	39	17	19	10
F	9	5	98	14	8	17	21	14
G	18	10	97	10	39	18	28	12

$0.02 \leq ADHcomplexity \leq 1$, and $0 \leq MFEcomplexity \leq 1$), the differences in the complexity values are puzzling.

In [7] we carried out similar experiments for the Bootstrapping and the Data splitting methods too. To compare visually the variability of the Random Subspace method with the other two, we display in Figures 3 to 6 the means for the 21 experiments (3 methods × 7 cases) and the minima and maxima as the error bars.

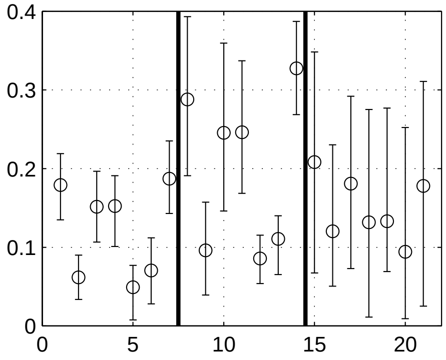

Figure 3: The mean and limits for $MSTcomplexity$. Bars 1-7: Bootstrapping, 8-14: Data Splitting, 15-21: Random Subspace

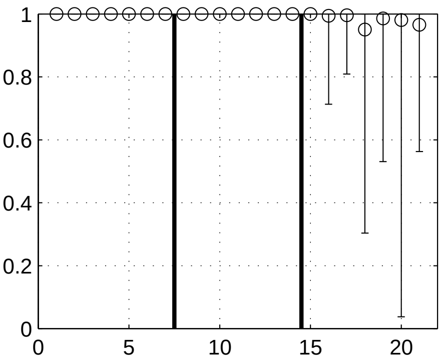

Figure 4: The mean and limits for *ADHcomplexity*. Bars 1-7: Bootstrapping, 8-14: Data Splitting, 15-21: Random Subspace

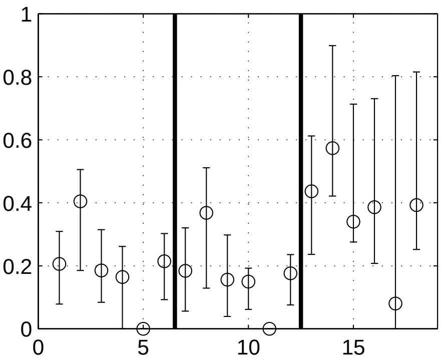

Figure 5: The mean and limits for *MFEcomplexity*. Bars 1-6: Bootstrapping, 7-12: Data Splitting, 13-18: Random Subspace

In all 4 figures the first 7 (six for MFE) bars are for the Bootstrapping method, the next 7 (6) are for the Data splitting method, and the last 7 (6) correspond with the Random Subspace method in the same order of the cases (A to G).

A common finding of all complexity measures is that the Random Subspace method for creating data subsets offers the highest variability of the complexity of the obtained sets. However, this seems to be the only finding where the four complexity measures agree. For example, while the ADH measure designates the Random Subspace method as producing the least complex data (Figure 4), the MFE measures classes these data set as the hardest (Figure 5).

The Bootstrap method has the lowest standard deviations (on all four measures) indicating that

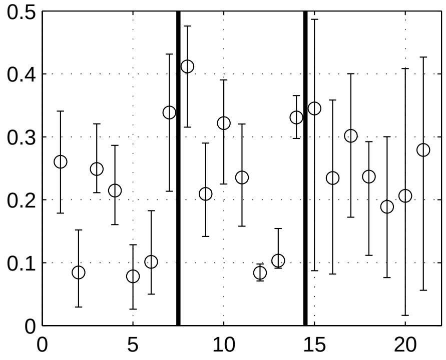

Figure 6: The mean and limits for *CLCcomplexity*. Bars 1-7: Bootstrapping, 8-14: Data Splitting, 15-21: Random Subspace

the data sets obtained exhibit complexity of a similar value.

These two findings can be explained by the following

1. The Bootstrap method creates data subsets by small variations of the original data set. Consequently, the variations in complexity among such data subsets can be expected to be small. In fact, as pointed out by Brieman [1], unstable classifiers are necessary to exploit effectively the low diversity of the data subsets generated by the Bootstrap method. Neural networks are examples of such unstable classifiers, and, curiously, we rely on their ability to overtrain.

2. Differently, the Random Subspace method usually generates data subsets exhibiting very different complexities because projecting the data set on a subspace may lead to a very different pattern of the classes' disposition. However, it should be noted that such variability in complexity strongly depends on the number of features used. It is easy to argue that variability in complexity among subsets should decrease as the number of randomly picked features increases. In addition, we think that complexity variations also depend on the degree of redundancy among the features. For example, if we have picked a subset of features containing redundant feature X_i, and another subset containing redundant feature X_j on its place, the difference in complexity between the two data sets will not be great. Ho conjectures in [3] that multiple classifier systems based on the Random Subspace method are expected to perform well exactly when there is a certain redundancy in the feature set. This comes in support of our intuitive hypothesis

that data sets of similar complexity (hence similar individual accuracy) are a better basis for a classifier combination system.

4.3 Additional Experiments on Random Subspace Method

In order to investigate how the complexity variations of Random Subspace depends on the number of random features used and on the degree of redundancy among the features, two experiments have been carried out:

1. We created subsets by randomly choosing 5, 7, 9, 11 of the 13 features of the wine data set. For each number of features, 50 subsets were created and the mean and the standard deviations were computed. We performed the experiments for all seven problems A–G.

2. We created a "redundant" version of the wine data set by adding 13 redundant features to the original feature space. The new 13 features were created by adding the first feature to each of the others, i.e., the new feature set consists of $\{X_1, \ldots, X_{13}, 2X_1, X_2 + X_1, \ldots, X_{13} + X_1\}$. For each of the seven problems A–G, 50 subsets were generated by randomly choosing 5 of the 13 features and the mean and the standard deviations were computed.

Figure 7 shows the behaviour of the four complexity measures as a function of the number of random features, for cases A to G. The means and the standard deviations are displayed. As it was hypothesized, the behaviour of the standard deviation shows that complexity variations among subsets decrease as the number of features increases whereas the mean tends to level off.

Table 4 shows the standard deviations obtained by sampling from the original feature space and the augmented feature space for the seven cases and the 4 complexity measures. The standard deviations for the augmented feature space are lower than the ones for the original feature space in most of the cases: for all 7 cases with MST; for 3 cases with ADH; for 5 cases with CLC; and 2 of the 6 possible cases with MFE. This points out that complexity variation among the generated data sets depends on the degree of redundancy as anticipated: the higher the redundancy, the smaller the variability.

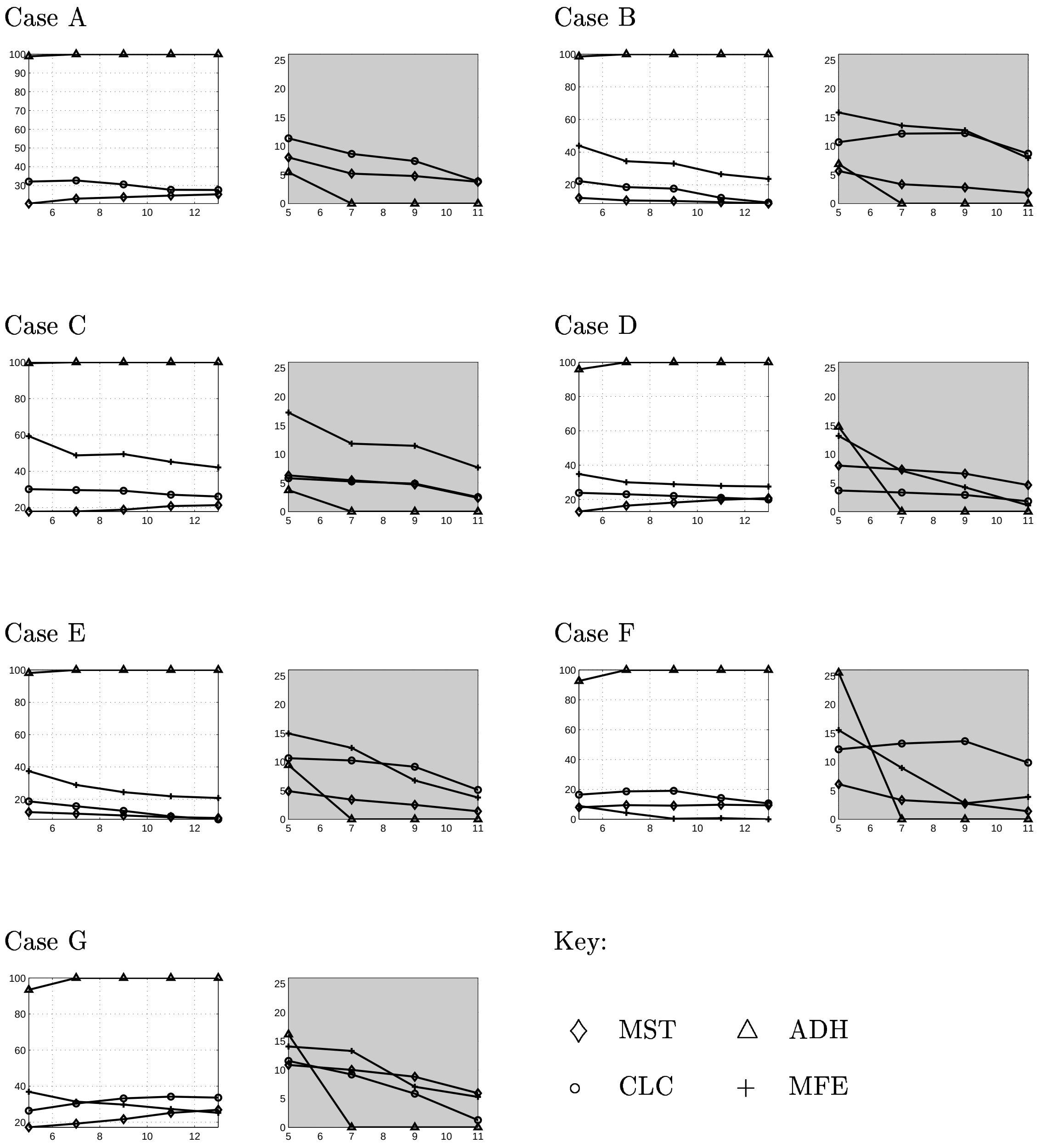

Figure 7: Plots of the means (left) and standard deviations (right, shaded) of the 4 complexity measures versus the number of features selected at random. 50 experiments have been carried out for each of the cases A-G for each of 5, 7, 9, and 11 features.

Table 4: Standard deviation (%) of the complexities for the Random Subspace method on the original and redundant feature spaces (augmented with redundant features).

Case	MST		ADH		MFE		CLC	
	orig	redn	orig	redn	orig	redn	orig	redn
A	8.04	4.35	5.45	12.05	11.37	5.68	–	–
B	5.68	2.12	6.88	9.21	10.69	6.85	15.86	13.67
C	6.30	3.48	3.73	8.95	5.81	8.39	17.24	18.61
D	8.01	3.73	14.77	11.40	3.68	4.64	13.17	17.35
E	4.90	3.01	9.44	10.94	10.63	3.42	14.94	14.43
F	6.12	4.50	25.52	21.89	12.19	9.85	15.53	27.80
G	10.86	6.41	16.17	14.42	11.55	8.06	14.06	18.95

5 Conclusions

In this paper, we considered the Random Subspace method for generating data sets for multiple classifier systems. To this end, we used 4 measures of complexity: the minimal spanning tree (MST), the adherence subsets measure (ADH), the maximal feature efficiency (MFE); and a cluster label consistency measure (CLC). Our results with the UCI "wine" data set led us to the following conclusions:

1. Random Subspace method usually generates data subsets exhibiting different complexities. The variability in complexity is higher than that for bootstrapping and data splitting (Figures 3 to 6). All 4 measures are capable of detecting this variablity, with ADH failing to distinguish between the complexity of the data sets for 7, 9 and 11 randomly selected features.

2. The variability in complexity of the data sets generated by the Random Subspace method is related to the number of features being selected. Our experiment showed that complexity variations among subsets decrease as the number of features increases whereas the mean complexity tends to level off (Figure 7). This complies with our hypothesis based on the idea that the more feature we use, the greater the chance for getting data sets on highly overlapping subspaces and hence the more similar the complexity.

3. The redundancy in the feature set *generally* leads to generating sets of more similar complexity compared to sets obtained from a feature set with little or no redundancy. While MST and CLC support this intuition (100 % for MST and ~70 % for CLC), ADH and MFE produce dubious results. According to the latter two measures, there is no clear pattern of reduction of the variability of the complexity value when redundant features are used.

Since there is no consensus on a single definition of complexity, we suggest that at this point we can use a (probably restricted) set of measures as a "complexity vector". This vector can be further used to select an appropriate classifier model for a certain data set or to indicate whether a collection of subsets is a suitable basis for a multiple classifier system.

References

[1] L. Brieman. Combining predictors. In A.J.C. Sharkey, editor, *Combining Artificial Neural Nets*, pages 31–50. Springer-Verlag, London, 1999.

[2] T.K. Ho. The random space method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

[3] T.K. Ho. Complexity of classification problems and comparative advantages of combined classifiers. In *Proc. First International Workshop on Multiple Classifier Systems*, pages 97–106, Cagliari, Italy, 2000.

[4] C. Ji and S. Ma. Combination of weak classifiers. *IEEE Transactions on Neural Networks*, 8(1):32–42, 1997.

[5] L.I. Kuncheva and C.J. Whitaker. Ten measures of diversity in classifier ensembles: limits for two classifiers.

[6] B. Littlewood and D.R. Miller. Conceptual modeling of coincident failures in multiversion software. *IEEE Transactions on Software Engineering*, 15(12):1596–1614, 1989.

[7] C.A. Shipp and L.I. Kuncheva. Four measures of data complexity for bootstrapping, splitting and feature sampling.