1

Data Complexity and Domains of Competence of Classifiers

Tin Kam Ho[1] and Ester Bernadó-Mansilla[2]

[1] Computing Sciences Research Center
Bell Laboratories, Lucent Technologies
Murray Hill, NJ 07974-0636 USA
e-mail: tkh@research.bell-labs.com
[2] Computer Engineering Department
Enginyeria i Arquitectura La Salle, Ramon Llull University
Quatre Camins, 2. 08022 Barcelona, Spain
e-mail: esterb@salleurl.edu

Summary. We study the domain of competence of a set of popular classifiers, by means of a methodology that relates the classifier's behavior to problem complexity. We find that the simplest classifiers, the nearest neighbor and the linear classifier, have extreme behavior in the sense that they mostly behave either as the best approach for certain types of problems or as the worst approach for other kinds of problems. We also identify that the domain of competence of the nearest neighbor is almost opposed to that of the linear classifier. Ensemble methods such as decision forests are not outstanding in any particular set of problems but perform more robustly in general. A by-product of this study is the identification of the most relevant features for optimal classifier selection.

Key words: Data complexity, domains of competence, classification.

1.1 Introduction

Research in machine learning and pattern recognition has yielded many competent classifiers from different families, such as decision trees, decision forests, support vector machines, neural networks, genetic algorithms, etc. Researchers have often demonstrated their competence and robustness across different domains. Nevertheless, the practitioner may find it difficult to choose a particular classifier for a given problem, due to the great variability of classifiers and a lack of knowledge on the optimal classifier for the given problem. Many classifiers appear in close rivalry in benchmark problems, which one can be selected? Many seem applicable to a wide range of problems, but will they also be suitable to the given problem?

The analysis of data complexity sets a framework to characterize the problem and identify domains of competence of classifiers. In [8] a methodology is introduced by which classification problems are characterized by a set of complexity measures.

This characterization allowed to identify easy problems (close to linearly separable problems) and difficult problems (close to random labeling) in the complexity measurement space. Derivations of this study led to relate the behavior of classifiers to problem complexity. The first attempt is made in [7], where two decision forests are compared to identify for which problems each is preferable. Last chapter studied the behavior of a particular classifier based on genetic algorithms called XCS. The study identifies the domain of competence of XCS compared with a set of other classifiers. In this paper, we extend this analysis to study the domain of competence of different classifiers.

We investigate the domain of competence of six popular classifiers in the complexity measurement space, and compare these domains to identify which classifiers are optimal for certain classes of problems. We include classifiers as diverse as a nearest neighbor, a linear classifier, an oblique decision tree, two types of decision forests, and XCS. We also analyze whether different classifiers have opposed domains of applicability or some of them perform similarly. Ensemble methods are shown to outperform single classifiers, but we aim to establish if single classifiers are still suitable to certain types of problems. Along this analysis we will validate the current measurement space and identify the set of complexity metrics most relevant for the identification of optimal classifiers.

This paper is structured as follows. First, we describe the methodology that we use to analyze the domain of competence of classifiers. Although this methodology is essentially the same as that described in the last chapter, we summarize it here to make the chapter self-contained. Section 1.3 analyzes where each classifier performs optimally and poorly. Section 1.4 takes a different view and analyzes the problems with a single dominant classifier and a single worst classifier. Section 1.5 discusses limitations of the current study and directions to overcome them, and finally, we summarize the paper and give the main conclusions.

1.2 Analysis Methodology

We characterize a classification problem by a set of complexity metrics. Table 1.1 summarizes the set of metrics used in our study. They are selected from [8] for being the best representatives of problem complexity. They describe different geometrical distributions of class boundaries, such as `boundary`, `intra-inter`, `nonlin-NN`, `nonlin-LP`, `pretop`, as well as the discriminant power of attributes (`fisher`, `max-eff` and `volume-overlap`). We include the ratio of the number of points to the number of dimensions (`npts-ndim`) as an estimation of sparsity. All these metrics are computed from the available training sets; therefore, they give measurements of the apparent complexity of problems.

We study the domain of competence of six classifiers:

- a nearest neighbor classifier (`nn`), with neighborhood set to 1 and Euclidian distance [1].
- a linear classifier (`lc`) computed by linear programming using the AMPL software [10].
- a decision tree (`odt`) using oblique hyperplanes [9]. The hyperplanes are derived using a simplified Fisher's method, as described in [6].
- a subspace decision forest (`pdfc`), which trains oblique trees on sampled feature subsets and combines them by averaging the posterior leaf probabilities [6].

Table 1.1. Complexity metrics used in this study

Measure	Description
`boundary`	percentage of points on boundary estimated by an MST
`intra-inter`	ratio of average intra-inter class nearest neighbor distances
`nonlin-NN`	nonlinearity of nearest neighbor
`nonlin-LP`	nonlinearity of linear classifier
`pretop`	percentage of points with maximal adherence subset retained
`fisher`	maximum Fisher's discriminant ratio
`max-eff`	maximum individual feature efficiency
`volume-overlap`	volume of overlap region of class bounding boxes
`npts-ndim`	ratio of the number of points to the number of dimensions

- a subsample decision forest (`bdfc`), also known as bagged decision trees, which trains oblique trees on sampled training subsets and then combines the result by averaging the posterior leaf probabilities [5].
- XCS, an evolutionary learning classifier [11, 12].

They have been selected for representing different families of classifiers. The nearest neighbor, the linear classifier and the single tree are traditional well-known classifiers. The forests belong to the category of classifier combination. They train several decision trees by subsampling either the training points (`bdfc`) or the features (`pdfc`). They are known to outperform single trees. XCS evolves a set of rules by means of a genetic algorithm. This particular selection allows to study if ensemble methods are always preferable than individual classifiers or on the contrary, there are still cases where single classifiers can be applied, and if so, where these cases are located in the measurement space. We do not pretend to have a fully representative set of classifiers. We rather want to try the methodology with an initial subset of classifiers and enhance this study to other popular classifiers once the methodology becomes mature.

We evaluate each classifier in a set of 392 two-class problems, extracted from the UCI repository [4], as explained in the previous chapter. For each problem, we estimate its complexity by computing each of the complexity metrics on the whole available dataset. We run each classifier using a ten-pass two-fold crossvalidation test and identify the best and worst classifier for each problem. Then, we compare each classifier against them respectively. Details are as follows:

1. Each dataset is randomly permuted ten times.
2. Each time, the dataset is divided in two disjoints sets. Then each classifier is trained in each of these two sets and tested on the other one. The classifier's error rate for this particular permutation is estimated as the sum of the errors on each test set, divided by the dataset size.
3. Thus, for each dataset there are ten error estimates, one for each permutation. The final classifier's error on the dataset is the average of its ten error rates.
4. For each problem, we identify the classifier with the lowest mean error. Then, we use its ten error estimates as the basis for comparison with the other classifiers, using a paired t-test with a 95% confidence level. Thus, we identify which classifiers are equivalent to the best method or worse than the best method.

5. The same procedure is used to identify the worst classifier for each problem and test the remaining classifiers against it, so that we identify classifiers equivalent to the worst method or better than the worst method.

We will approach the domains of competence of classifiers from two different views. The first one, taken in section 1.3, will estimate the domain of competence of each classifier. We analyze where each classifier performs as the best method, and as the worst method, trying to identify types of problems where the classifier is well suited and poorly suited. The second approach, taken in section 1.4, analyzes, for each problem, the set of classifiers that are well suited and poorly suited. Although the views are similar, here we will distinguish the problems where there is a single dominant best classifier, from problems where more that one classifiers are optimal. Thus, we will try to determine if there are differences between these types of problems. We will also study the problems where a single classifier performs significantly worst from those problems where several classifiers are equivalently poor. This approach was already taken in [3] where the domains of dominant competence were succinctly identified. This study is enhanced here with the addition of the first approach. Both views are necessary to help choose an optimal classifier given a problem with computed complexity metrics. The results obtained herein are tied to the particular choice of classifiers, so other choices or the inclusion of new classifiers may lead to different results.

1.3 On the Domains of Competence of Classifiers

We analyze where each classifier performs as one of the best methods and as one of the worst methods. We try to relate this to the complexity measurement space so that we can identify domains of competence of classifiers.

We will show different projections of the complexity measurement space and plot the classifier's membership to three categories: best, worst, or none of them. We use a circle when the classifier is equivalent to the best method (i.e., it is the best method or its performance is equivalent to the best method considering a paired t-test with a 95% confidence level). We use a cross when the classifier is equivalent to the worst method (which means it is the worst classifier or its performance is found equivalent to that of the worst classifier). The rest of the problems, where the classifier is neither best nor worst, are shown with a small plus sign.

Nearest Neighbor

Fig. 1.1 shows the domain of competence of the nearest neighbor classifier. We find that most of the problems where `nn` performs optimally belong to very low percentage of points in boundary and low nonlinearities (see Fig. 1(a)). They also tend to be placed in low intra-interclass nearest neighbor distances, as shown in Fig. 1(b), although other problems with low values in this metric do not correspond to an optimal `nn`'s behavior. The remaining metrics do not influence the classifier's behavior significantly. The pretopological measure on the percentage of retained adherence subsets is not very significant to set the `nn`'s behavior, as it is shown in Fig. 1(c). The discriminant power of the attributes is neither significant for determining the `nn`'s behavior; see for example the maximum's Fisher discriminant ratio in Fig.

Table 1.2. Percentage of problems where each classifier is equivalent to the best method, equivalent to the worst method, and none of them.

	Best(%)	Worst (%)	Avg. (%)
`nn`	54	34	12
`lc`	33	40	27
`odt`	3	70	27
`pdfc`	12	10	78
`bdfc`	20	11	69
`xcs`	19	17	64

1(b). Analyzing `npts-ndim` it seems that almost all the problems where the `nn` is optimal correspond to high ratios of `npts-ndim` (about 100). But this is just a coincidence because almost all problems located in this value belong to the letter problem, which all have the same relation of the number of points over the number of dimensions. The problems do not present a uniform distribution over this metric, so it is difficult to extrapolate observations from it. We note that the `nn` is optimal for the easiest problems; observe that in these cases, the `nn`'s error is very low. We also verified that these problems also correspond to low errors from the remaining classifiers.

Table 1.2 summarizes the number of problems where each classifier is best, worst and average. We identify that the nearest neighbor has the extreme behavior of either behaving mostly like the best classifier (54% of the problems) or like the worst classifier (34% of the problems). There is a greater tendency to behave optimally although this may be biased by the current selection of problems. Only in 12% of the problems, the `nn` is an average classifier.

Linear Classifier

The linear classifier has a behavior almost contrary to that of the nearest neighbor. Fig. 1.2 shows that for very low percentage of points in boundary (less than 10%) it performs as the worst method, while it works as the best method for a number of problems with `boundary` values between 10% and 70%. Nevertheless, there are also some few problems with percentage of points in boundary inside this range (10%-70%) where the linear classifier performs as the worst method or as an average method. Tracing the `lc`'s behavior along the different projections of complexity, we identify that the `lc` performs best for high boundary values, high intra-interclass nearest neighbor distances, and high nonlinearities of both the linear classifier and nearest neighbor. But there are also some problems (although in fewer proportions) that are placed in similar regions of the measurement space, where the linear classifier performs as the worst method.

The general tendency is that the linear classifier performs optimally when the problems are more difficult. For very easy problems (few points in boundary, low nonlinearities, etc) the linear classifier, although having a low error, is the worst method.

This behavior is in fact surprising; one can question how a linear classifier separating the class boundaries by an hyperplane can overcome other more sophisticated classifiers as the decision forests or XCS, specially in the most difficult problems. We

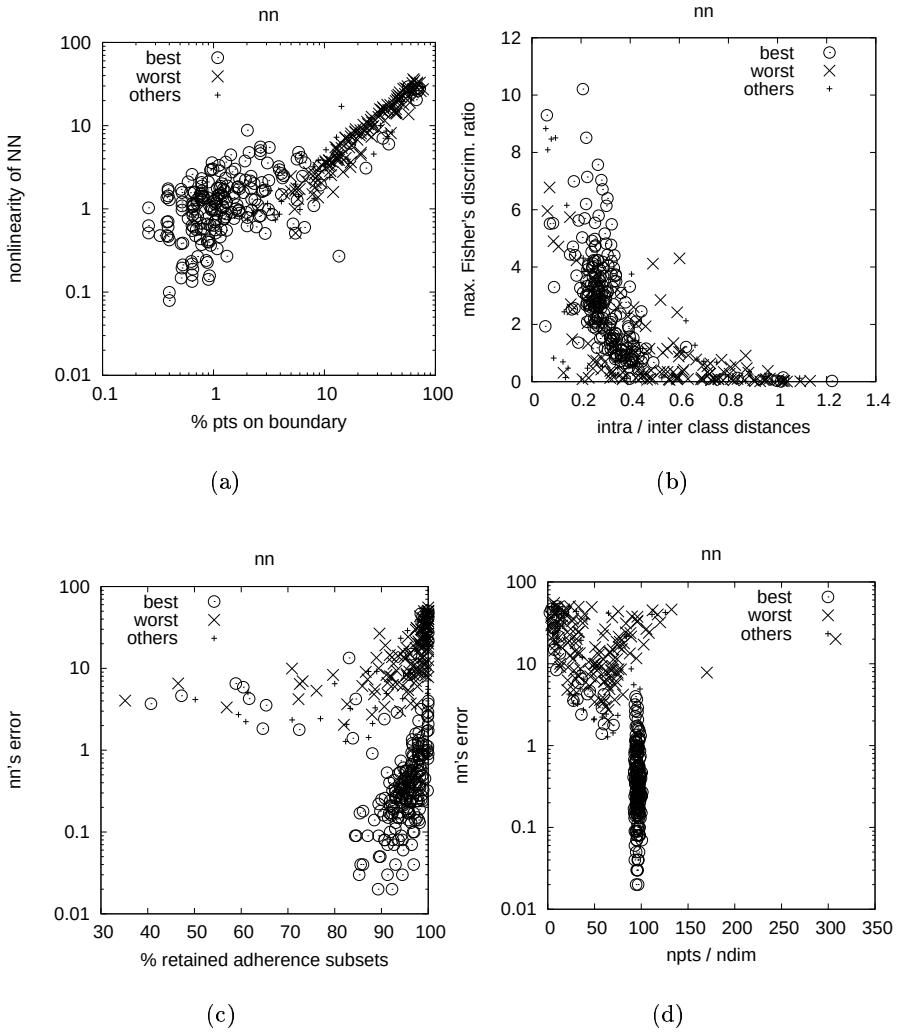

Fig. 1.1. Problems where the nearest neighbor (nn) performs best ($\odot$), worst ($\times$), and average ($+$), shown in selected projections of the complexity measurement space: (a) percentage of points in boundary vs. nonlinearity of nn, both in logarithmic scale, (b) intra-interclass nearest neighbor distances vs. maximum Fisher's discriminant ratio, (c) percentage of retained adherence subsets vs. nn's error (in logarithmic scale), and (d) ratio of the number of points to the number of dimensions vs. nn's error (in logarithmic scale).

hypothesize that sparsity of the training set may be a possible cause, making other classifiers overfit. In sparse training sets, sophisticated classifiers may try to approximate too precisely the class boundaries when these boundaries are not described by sufficient representative points. Then, these classifiers may perform poorly with new unseen instances. A linear approach may be well suited to this type of problem, having less tendency to overfit. The number of points to the number of dimensions tries to approximate the sparsity of the training set. But we find no clear relation between the `lc`'s behavior and this metric, which may indicate that the metric is a rough estimation of the training set sparsity. In fact, the metric can only consider the apparent sparsity of the training set, which may be uncorrelated from the true sparsity. The distribution of points in the available training set might be very different from the original distribution of the problem. As we do not have the original sources of the datasets, we can not compute the true sparsity for the current set of problems.

Since we find that the linear classifier tends to behave optimally for the most complex problems, where the classifiers' errors are very high, we may also hypothesize that this condition can be due to the presence of noise, i.e., the presence of mislabeled points in these datasets. For these types of problems, a linear classifier may be more robust than other classifiers that try to evolve more complex boundaries, which result to be too overfitted.

The current measurement space is insufficient to distinguish clearly the problems where the linear classifier is best and worst. Although the reasons may be justified by the two previous hypotheses, they may also be due to a lack of metrics describing more precisely the complexity of the problem.

The `lc`'s behavior is somehow extreme too, as found in the case of the `nn` classifier. Observe that `lc` is optimal in 33% of cases, and worst in 40% of cases, as shown in Table 1.2.

Decision Tree

The decision tree performs optimally in very few problems; to be exact, in only 12 problems out of 392, which corresponds to a percentage of 3%. In 70% of the problems, the single tree performs as the worst method, while it performs in the average in 27% of the problems, as shown in Table 1.2.

It is difficult to determine for what kind of problems the single tree is best suited. It only performs optimally in 12 problems, which is not sufficient to extrapolate general observations. Moreover these problems are not compacted in the same area of the measurement space, as it can be observed from Fig. 1.3. It is also difficult to discriminate between the problems where the single tree performs worst from the problems where the single tree is an average performer.

Subspace Decision Forest

The subspace decision forest improves the behavior of the single tree, in the sense that the forest is more robust in a high proportion of problems. Table 1.2 shows that the subspace forest is an average method in 307 problems (78%), and is best and worst in fewer proportions (12% and 10% respectively).

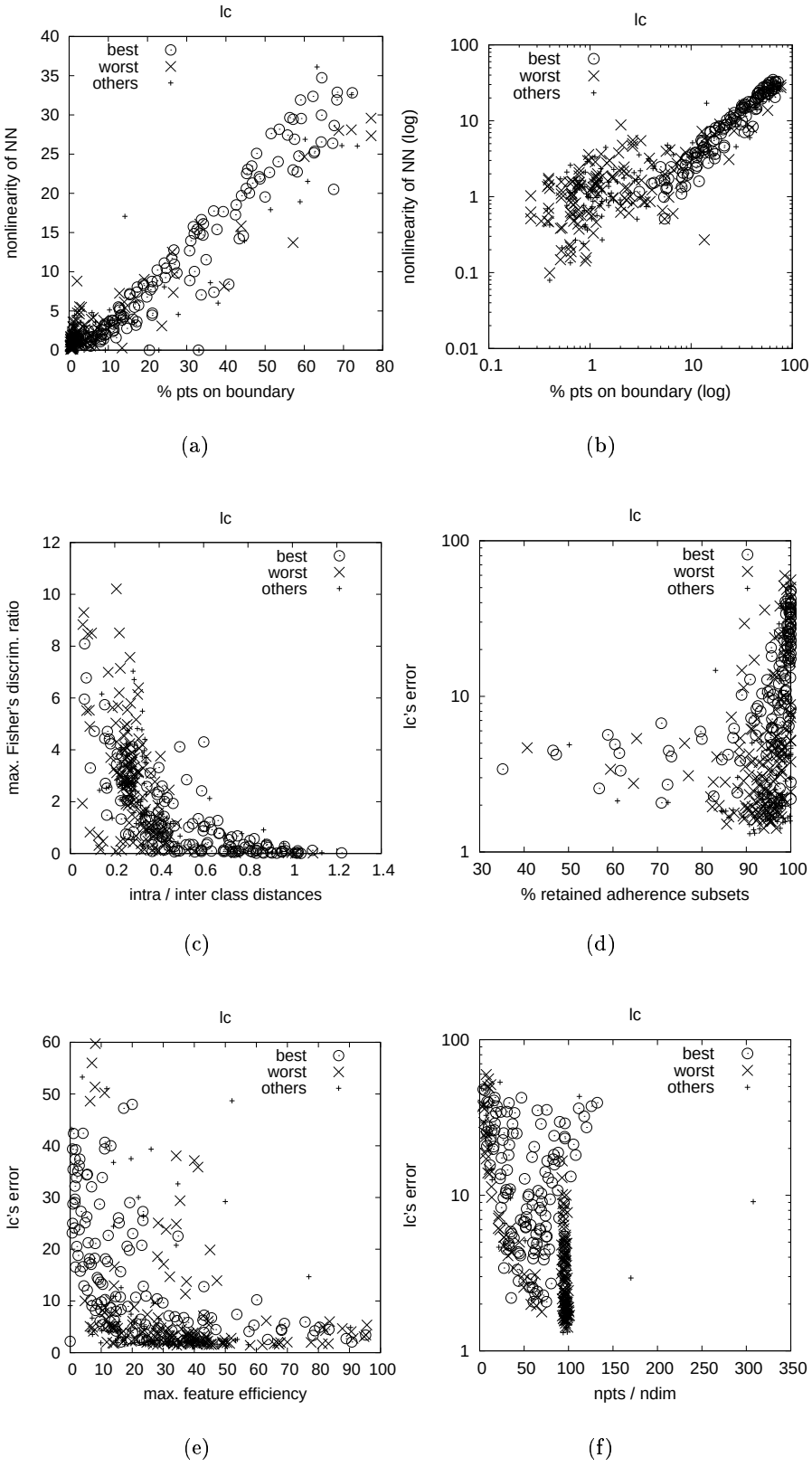

Fig. 1.2. Problems where the linear classifier (`lc`) performs best (⊙), worst (×), and average (+), shown in selected projections of the complexity measurement space: (a) percentage of points in boundary vs. nonlinearity of `nn`, (b) percentage of points in boundary vs. nonlinearity of `nn`, in logarithmic scale, (c) intra-interclass nn distances vs. maximum Fisher's discriminant ratio, (d) percentage of retained adherence subsets vs. `lc`'s error, (e) maximum individual feature efficiency vs. `lc`'s error, and (f) ratio of the number of points to the number of dimensions vs. `lc`'s error.

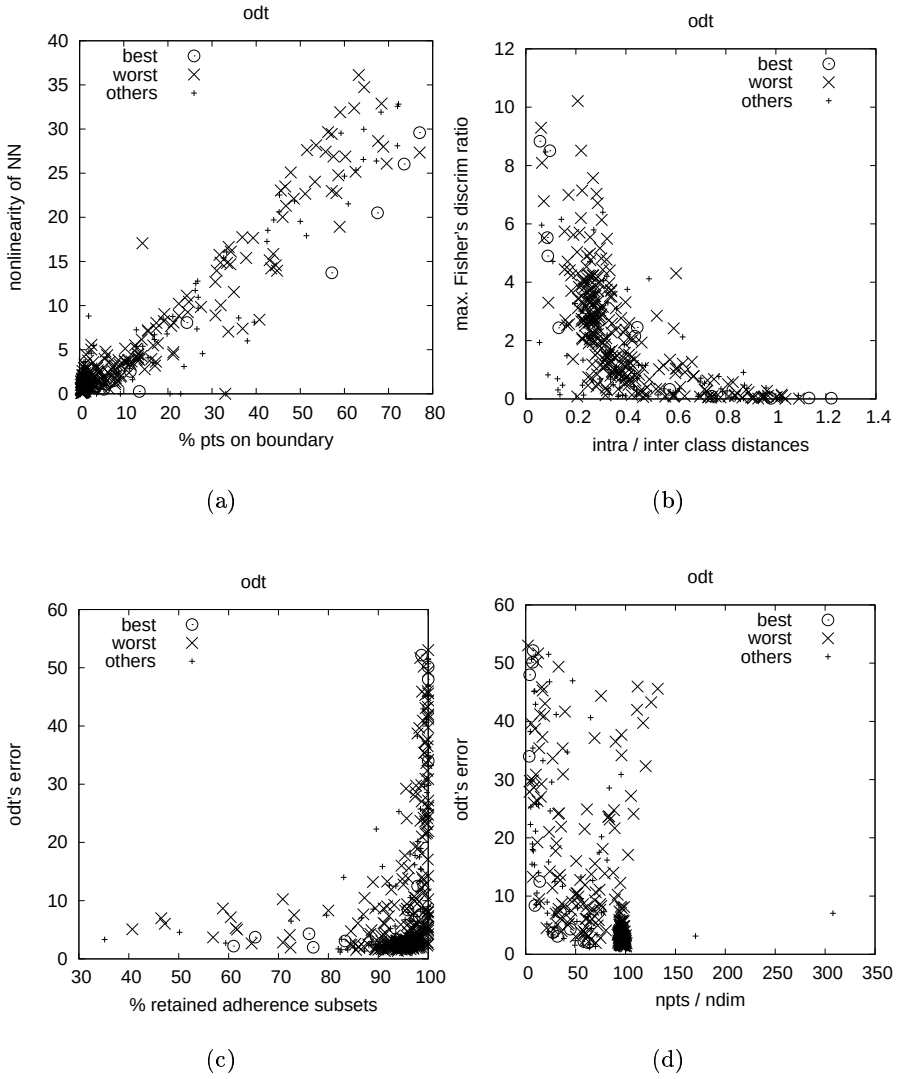

Fig. 1.3. Problems where the single tree (`odt`) performs best ($\odot$), worst ($\times$), and average ($+$), shown in selected projections of the complexity measurement space: (a) percentage of points in boundary vs. nonlinearity of `nn`, (b) intra-interclass nearest neighbor distances vs. maximum Fisher's discriminant ratio, (c) percentage of retained adherence subsets vs. `odt`'s error, and (d) ratio of the number of points to the number of dimensions vs. `odt`'s error.

Fig. 1.4 shows the `pdfc`'s behavior on selected projections of the complexity measurement space. The plots do not show very compact areas of the measurement space to distinguish clearly between the best, average and worst problems for the subspace decision forest. But the general trend is that a higher percentage of problems where the `pdfc` performs optimally belong to small percentage of points in boundary, and also small nonlinearities and intra-interclass nearest neighbor distances. However, observe that for very few points in boundary and small nonlinearities, the best classifier is the nearest neighbor, as shown in Fig. 1.3. For `boundary` values higher than 30% there are cases where the `pdfc` is best and also other cases where the `pdfc` is worst or average. Other projections of the complexity measurement space do not show any significant discrimination between these three cases.

Subsample Decision Forest

Comparing the subsample forest with the single tree, we find that the subsample forest has more robustness across a high range of problems. In almost 69% of the problems, the subsample decision forest is in the average, being as the best method in 20% of problems and worst in the remaining 11% (see Table 1.2). A similar behavior in terms of robustness is observed with the subspace decision forest.

Nevertheless, it seems that the subspace decision forest and the subsample decision forest do have differences in their domains of competence. Comparing Fig. 1.4 with Fig. 1.5, we note that the subsample forest is able to be optimal in problems with higher `boundary` values than the subspace forest. In fact, the average percentage of points in boundary is 14.38% for the problems where the subspace forest is best, and 30.37% for the problems where the subsample forest is best. The same behavior is observed with the nonlinearities. While the subspace forests work best for low nonlinearities, the subsample forests work best for higher values. This also happens with the ratio of intra-interclass nearest neighbor distances. These results are consistent with previous experiments in the literature [7], where subspace decision forests are compared with subsample decision forests.

XCS

The domain of competence of XCS is already analyzed in the previous chapter. Summarizing our results, we found that XCS performs best for low points in boundary. For the lowest percentages of points in boundary, XCS is in the average methods (in these cases, the best classifier is the nearest neighbor). XCS also tends to be optimal for low nonlinearities and low ratios of intra-interclass nn distances. The maximum's Fisher discriminant ratio and the number of points to the number of dimensions tend also to be higher for the problems where XCS is best.

The domain of competence of XCS appears to have similarities with that of the subspace decision forest (see [2]). In fact, we can view XCS as a type of classifier ensemble method, where each classifier contains a rule with generalizations in some attributes, having an effect similar to that of sampling over the feature space.

Comparative Analysis

Fig. 1.6 compares jointly the domains of competence of each classifier against some selected metrics. Each column refers to a classifier; from left to right these are

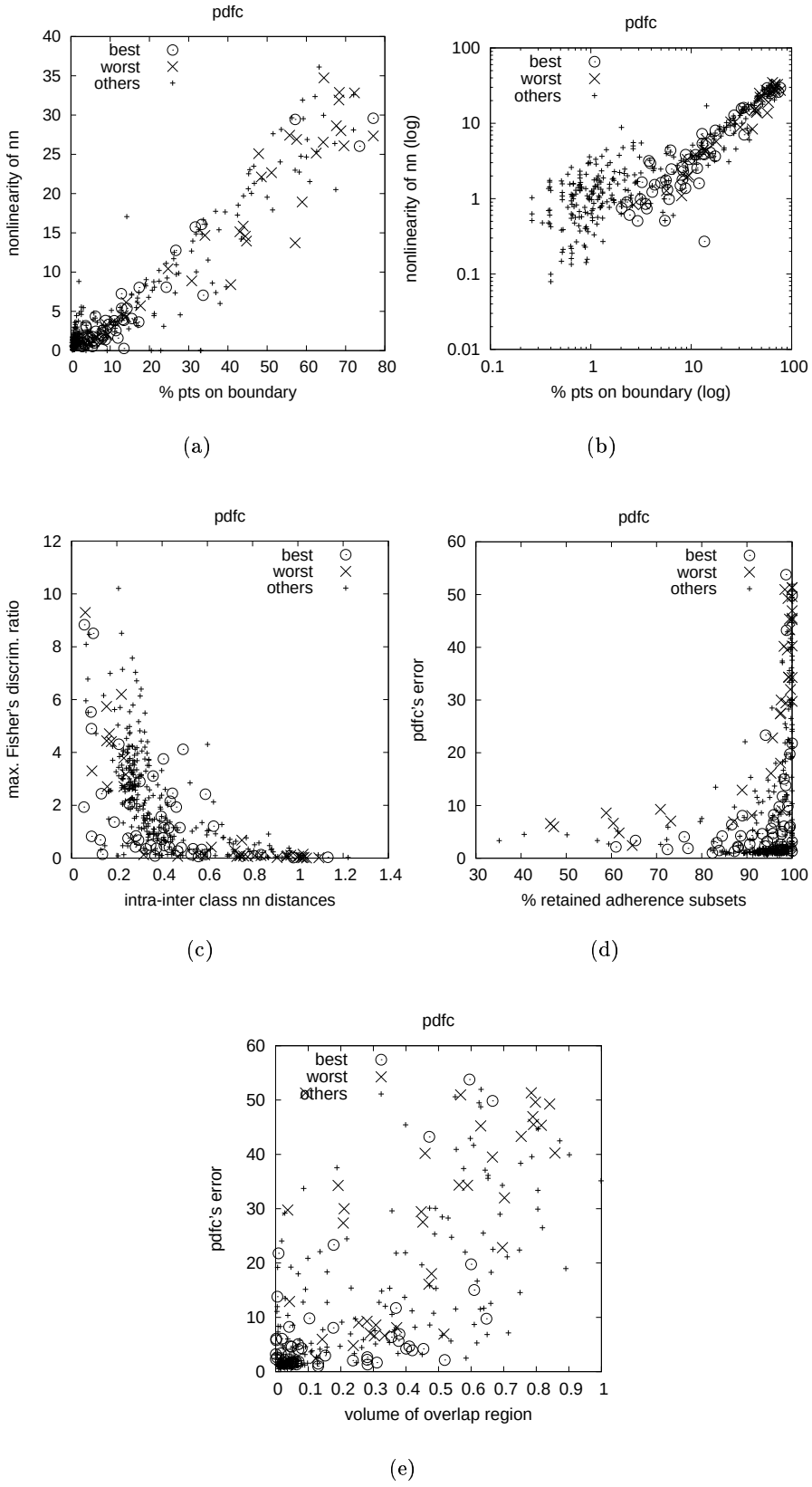

Fig. 1.4. Problems where the subspace decision forest (`pdfc`) performs best ($\odot$), worst ($\times$), and average ($+$), shown in selected projections of the complexity measurement space: (a) percentage of points in boundary vs. nonlinearity of `nn`, (b) percentage of points in boundary vs. nonlinearity of `nn` in logarithmic scale, (c) intra-interclass nearest neighbor distances vs. maximum Fisher's discriminant ratio, (d) percentage of retained adherence subsets vs. `pdfc`'s error, and (e) volume of overlap region vs. `pdfc`'s error.

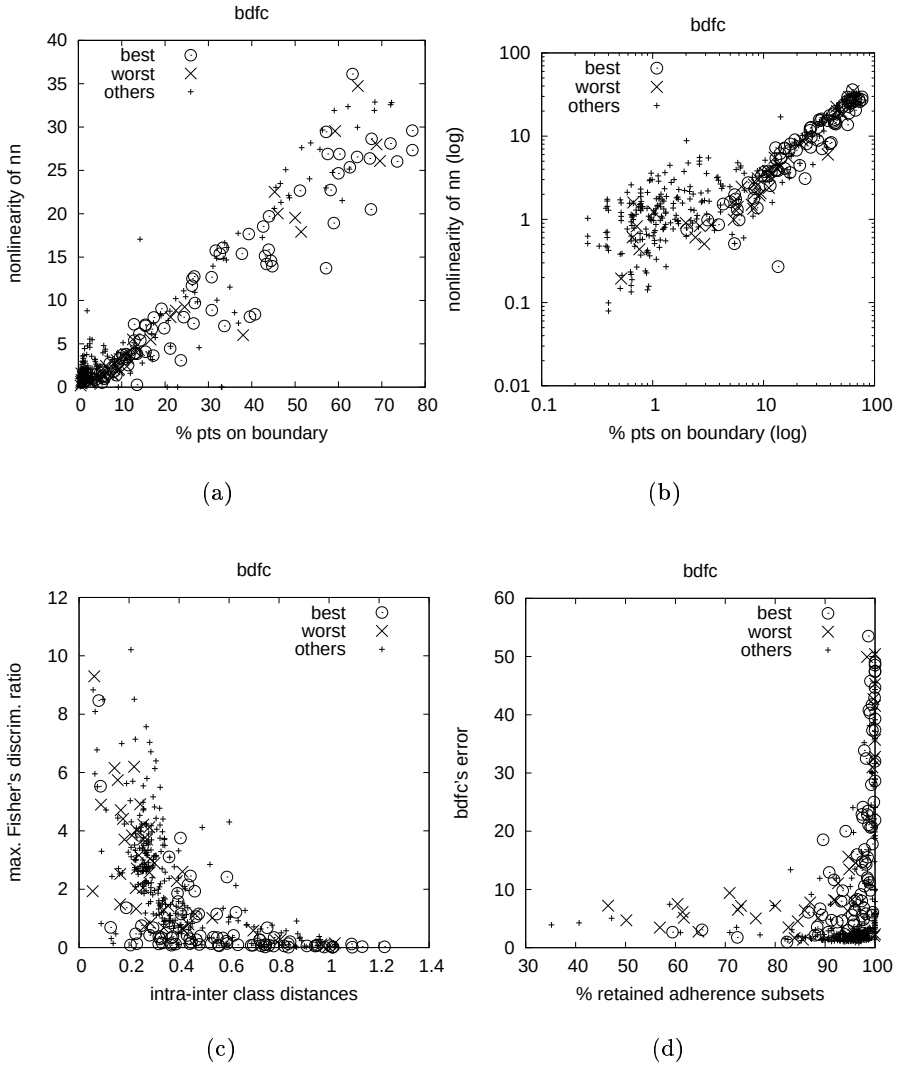

Fig. 1.5. Problems where the subsample decision forest (`bdfc`) performs best ($\odot$), worst ($\times$), and average ($+$), shown in selected projections of the complexity measurement space: (a) percentage of points in boundary vs. nonlinearity of `nn`, (b) percentage of points in boundary vs. nonlinearity of `nn`, in logarithmic scale, (c) intra-interclass nearest neighbor distances vs. maximum Fisher's discriminant ratio, and (d) percentage of retained adherence subsets vs. `bdfc`'s error.

the `nn`, `lc`, `pdfc`, `bdfc`, and `XCS`. The single tree is omitted because of its scarce contribution as an optimal classifier. Each row plots a particular complexity metric. Each figure shows three boxplots summarizing the complexity distribution of each classifier when it performs best (1), worst (2), or average (3). The boxplot has a box with lines at the lower quartile, the median and the upper quartile. Whiskers extend to 1.5 times the box length, and the remaining points are considered as outliers and are plotted with points. The boxplot is useful to analyze the distribution of each type of problems succinctly, because the ranges and the spread of data can be easily observed. Nevertheless, the number of points in each boxplot remains hidden so that this must be coupled with the previous figures.

Note that the comparison of complexity distributions between the `nn` and the `lc` emphasizes again that these classifiers have opposed domains of competence, as seen specially in measures such as `boundary`, `nonlin-NN`, `intra-inter`, and `volume-overlap`. The `fisher` metric is not as relevant, although we also observe a tendency for higher discriminant attributes in problems where the `nn` is best. The decision forests have different domains of competence, being the measures related with class distributions, `boundary`, `nonlin-NN` and `intra-inter`, the most discriminant ones. XCS's domain of competence appears again very similar to that of the subspace decision forest. Again the three metrics `boundary`, `nonlinNN` and `intra-inter` show high correlations for the domains where XCS and the subspace decision forests are best and worst.

1.4 Dominant Competence of Classifiers

So far we have studied the problems where each classifier performs best and worst. The approach taken was that of analyzing each classifier separately and relate the results to the complexity measurement space. In this section, we take a different point of view. We analyze, for each problem, which is the best and worst classifier solving it. Doing this, we have observed that some problems are solved by only one dominant method. On the contrary, other problems have more than one outstanding methods. These are problems where several classifiers can obtain good results and therefore their study is less important. Therefore, we will focus on the first types of problems, that is, problems that are only solved by one dominant method. A similar behavior is observed for the worst methods of each problem. Some problems have only one worst classifier, while others have more than one worst classifiers. We will also analyze which kinds of problems present only one worst classifier.

There are 270 problems that have a dominant best classifier. This represents a 69% of all the datasets. Fig. 1.7 shows these problems, plotted against selected projections of the measurement space. We use a different symbol for each classifier, as indicated in the legend of each plot. We also show with small dots the location of the problems with more than one optimal classifiers. Note that there are only four methods which are dominant out of six methods. These are the nearest neighbor, the linear classifier, the subsample decision forest, and XCS. The rest of classifiers (the single tree and the subspace decision forest) are not outstanding in any problem. When they perform as the best method, there are also other methods performing equivalently.

Moreover, it is also interesting to note that almost all these problems are solved predominantly by the nearest neighbor or the linear classifier. Table 1.3 details the

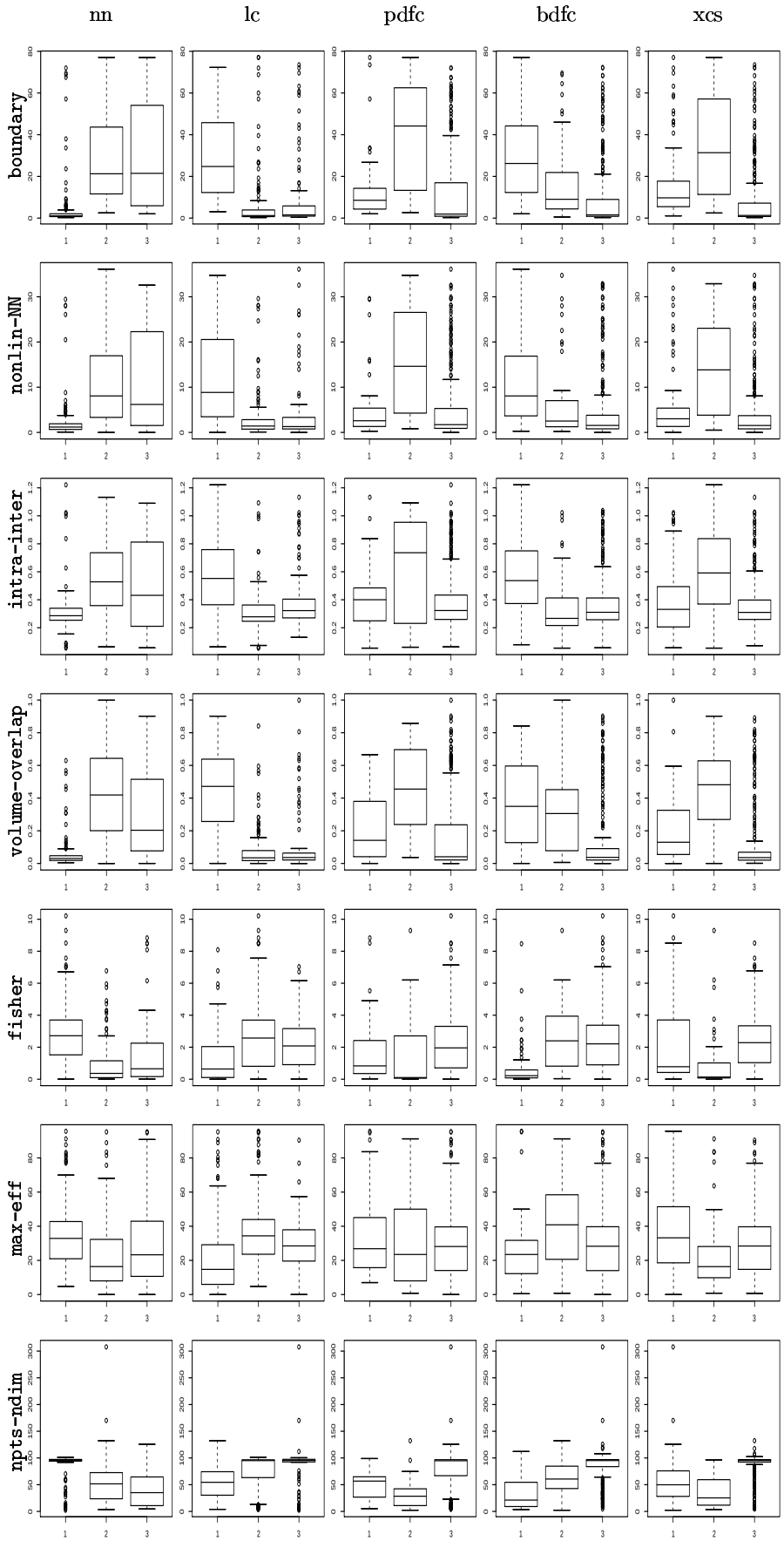

Fig. 1.6. Boxplot distributions of best (1), worst (2), and average (3) domains for each classifier, shown in individual projections of the complexity measurement space.

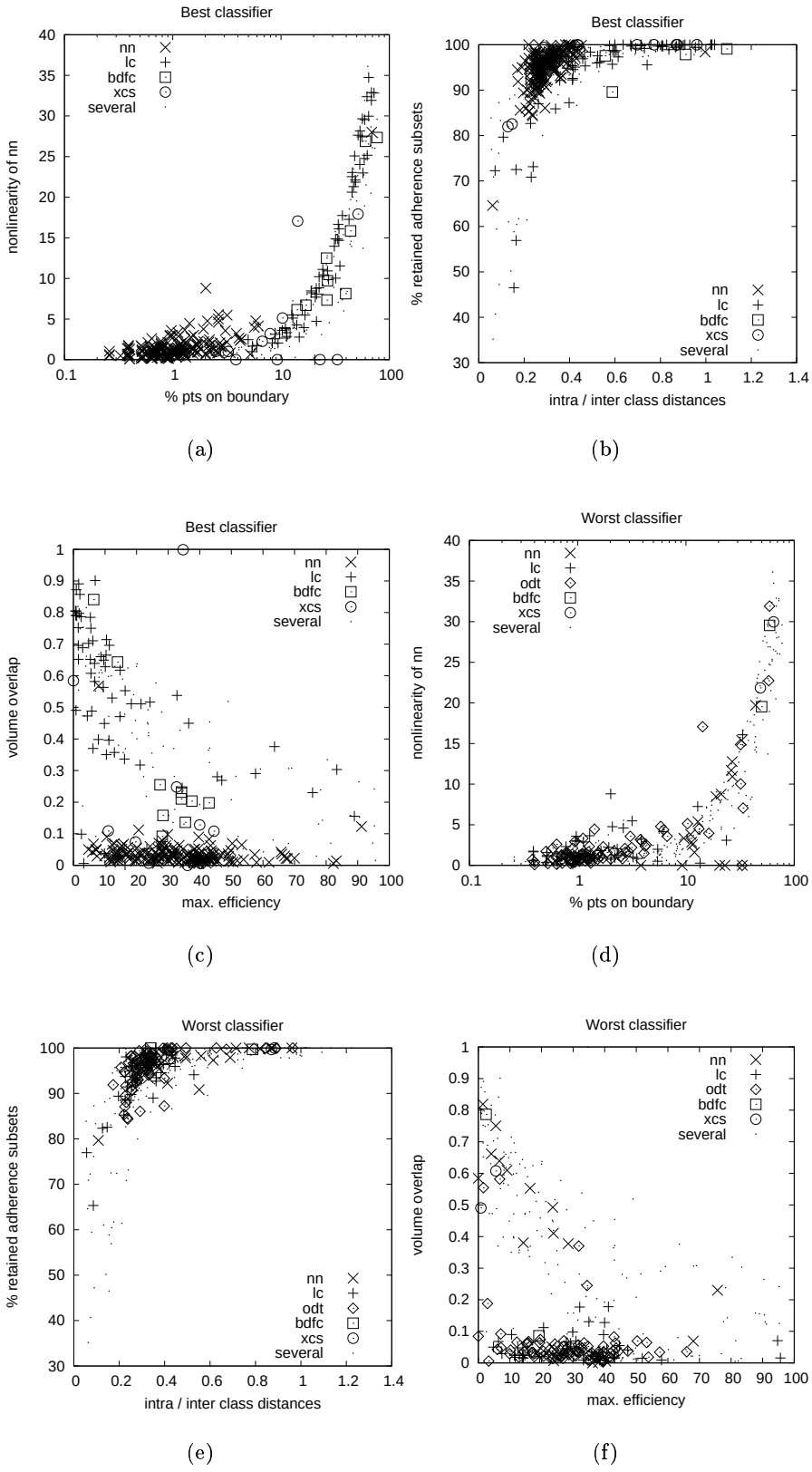

Fig. 1.7. Dominant competence of classifiers. Problems with a dominant best classifier (a, b, and c), and problems with a dominant worst classifier (d, e, and f), shown in selected projections of the complexity measurement space: percentage of points in boundary vs. nonlinearity of nn (a and d), intra-interclass nearest neighbor distances vs. percentage of retained adherence subsets (b and e), and maximum feature efficiency vs. volume of overlap region (c and f).

Table 1.3. Distribution of classifiers for problems with a best dominant classifier and a worst dominant classifier.

	Best	Worst
nn	69%	11%
lc	23%	30%
odt	0%	56%
pdfc	0%	0%
bdfc	4%	1%
xcs	4%	1%
total	270	157

proportion of problems where each method is predominantly best and worst. See that the `nn` classifier is dominantly best in 69% of problems, and worst in 11%. The `lc` is dominantly best in 23% of problems and worst in 30%. This is significantly different from the forests and XCS; they are almost neither dominantly best nor worst. We can conclude that the nearest neighbor and the linear classifier are very specialized methods, being successful only for specific types of problems. On the other hand, the ensemble methods and XCS are more robust but they are not outstanding in many problems. Fig. 1.7 also shows that the domain of competence of the nearest neighbor is placed in low percentage of points in boundary, and low nonlinearities. For increasing `boundary` values, XCS seems to be the best classifier, although for a small range of problems (in `boundary` values between 2% and 10%). For higher `boundary` values, there is a range of problems where the linear classifier mostly outstands, but also sometimes, and with less frequency, the subsample forest and XCS.

It is also interesting to compare Figs. 7(a), 7(b), and 7(c) with Figs. 7(d), 7(e) and 7(f), which show the problems where there is only one method performing poorly. There are 157 problems with a single worst classifier. Note that the single tree appears very often as the dominant worst classifier, mainly located in low `boundary` values. Also the linear classifier is worst for low `boundary` values and low nonlinearities. The nearest neighbor tends to be worst for percentage of points in boundary greater than 10% and nonlinearities greater than 4% approximately. XCS appears as the worst method for high boundaries and nonlinearities.

We also note that there is no compact area where problems solved by more than one outstanding (best or worst) methods are placed. They are distributed along all projections of the complexity measurement space, so we can not give any apparent reason to discriminate problems with a dominant method from those with several applicable methods, at least for the current measurement space.

1.5 Discussion

The current study allows also to identify the most relevant metrics for discriminating between domains of competence of classifiers. These are: the percentage of points in boundary, the nonlinearities, and the ratio of intra-interclass nearest neighbor distances. These metrics are more related to the geometry of the problem and the shape and distribution of the class boundaries. The metrics describing the discrim-

inative power of the attributes, like the maximum Fisher's discriminant ratio and the maximum feature efficiency, seem to be less important to identify domains of competence. Although they influence the complexity of the problem, they are not as useful as the other metrics to discriminate between two classifiers. This means that the domains of competence of classifiers are mostly determined by the geometry of the problem.

Some metrics, although also describing the geometry of the problem, are able to give particular explanations of some classifier's behavior, but they are not general enough. One of the reasons of their narrower applicability is that they are not spread uniformly for the current set of problems. This is the case of the pretopological measure on the percentage of retained adherence subsets. It presents high values for almost all the problems (between 80% and 100%). Although small values may indicate that the problem has a less complex geometry, there are insufficient problems located in these cases to extract useful conclusions. Moreover, there are empty regions in the measurement space. The problems are not evenly distributed along all dimensions of the measurement space; i.e., there are some regions which are not covered by any problem. We still do not know if these empty regions are induced by some geometrical constraints or are due to the particular choice of classification problems.

Another source of difficulty for the current study, which limits the extraction of more conclusive results, is the estimation of the complexity of the problems. Recall that all metrics are computed from the available training sets, and therefore they represent the apparent complexity of the problem. The measure of sparsity seems to be particularly sensitive to it. The estimation of the sparsity of the training set by the ratio of the number of points to the number of dimensions on the available training set may be uncorrelated with the real distribution of points in the original problem. This leads us to inconclusive results when we try to explain the domain of competence of the linear classifier related to the sparsity of the training set.

On the other hand, there are also some correlations between the metrics themselves. For example, the percentage of points in boundary is fairly related to the nonlinearities, and to the intra-interclass nearest neighbor distances to some extent. Although this correlation is reasonable, it is not necessary. For example, a problem can have a high percentage of points in boundary but present low nonlinearities. The correlation between these metrics in the current set of problems may lead to conclusions too overfitted for these problems.

The lack of uniformity, the apparent estimation of metrics and the correlation between metrics are some of the sources of difficulty found in the present study. Also the current choice of problems may bias the results and lead to conclusions which may not be directly extrapolated to other types of problems. The use of problems designed artificially may overcome these difficulties.

1.6 Conclusions

We propose a methodology based on the analysis of data complexity to study the domains of competence of classifiers. We find that the simplest methods, i.e., the linear classifier and the nearest neighbor, have *extreme* behaviors. They perform optimally in a number of problems (third part and half respectively), but also perform as the worst method in almost the same percentages of problems respectively. This

means that they are very specialized methods. When the conditions are favorable for these particular methods, they perform optimally. The key issue is to detect which conditions are these and whether they apply given a certain problem. The single decision tree is almost biased towards performing as the worst method or as an average method. On the other hand, the most elaborate classifiers tend to have a more robust behavior. They are mostly placed between the best and the worst method. They are not very specific to behave optimally in a particular set of problems, nor to behave poorly in another type of problems, but to behave in the average for a high proportion of problems. These types of classifiers are more general methods. They are applicable to a higher range of problems, where we can expect a moderately good result. This happens with the subsample decision forest, the subspace decision forest, and XCS.

The key issue is to identify which type of problems are suitable and not suitable for each classifier. This is more important for the most specialized classifiers, since their behavior can change dramatically depending on the problem. The domain of competence of the nearest neighbor classifier is located in problems with compact classes and few interleaving. Particularly, for problems with less than 10% of points in boundary, intra-interclass nearest neighbor distances less than 0.4 and nonlinearities less than 5%, the nearest neighbor classifier has good applicability. For problems outside this region, the nearest neighbor classifier is hardly recommended. The identification of the domain of competence of the linear classifier is more difficult. We effectively identify that the linear classifier is not well suited for problems with compact classes. In these cases, other classifiers perform better. But to what kind of problems the linear classifier is best suited is not conclusive enough, at least for the current measurement space. A possible hypothesis points out to problems with sparse training sets, but this is also difficult to determine since we do not know the original distribution of the problems. The decision tree is almost always outperformed by the other classifiers. Nevertheless, the ensemble classifiers based on the same tree, the subspace decision forest and the subsample decision forest, are much more applicable. In fact, the ensemble classifiers and XCS are average methods for a wide range of problems. In cases of uncertainty, so that there is no guarantee that a simple classifier will perform best, ensemble methods can offer a reasonable result. In these cases, XCS seems to perform better when the classes are more compact, similarly to the subspace decision forest. On the contrary, the subsample decision forest works better for higher percentage of points in boundary and higher nonlinearities.

Limitations of the current study are identified, such as biases due to the current choice of problems, uneven distribution of problems along the measurement space, and apparent estimation of complexity. The study is also biased by the current pool of classifiers. Further studies are needed to provide higher understanding on the relationship between class distributions, complexity and classifier's behavior. Also adding in synthetic datasets may be useful to control the apparent estimation of complexity and its influence on data complexity.

Acknowledgement. Ester thanks the support of *Enginyeria i Arquitectura La Salle*, Ramon Llull University, as well as the support of *Ministerio de Ciencia y Tecnología* under project TIC2002-04036-C05-03, and *Generalitat de Catalunya* under Grant 2002SGR-00155.

References

1. D.W. Aha, D. Kibler, and M.K. Albert. Instance-based Learning Algorithms. *Machine Learning, Vol. 6*, pages 37–66, 1991.
2. Ester Bernadó-Mansilla and Tin K. Ho. Domain of Competence of XCS Classifier System in Complexity Measurement Space. *IEEE Transactions on Evolutionary Computation*, 9(1):82–104, 2005.
3. Ester Bernadó-Mansilla and Tin Kam Ho. On Classifier Domains of Competence. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 1, pages 136–139, 2004.
4. C.L. Blake and C.J. Merz. UCI Repository of machine learning databases, [http://www.ics.uci.edu/∼mlearn/MLRepository.html]. University of California, Irvine, Department of Information and Computer Sciences, 1998.
5. L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
6. Tin K. Ho. The random subspace method for constructing decision forests. *IEEE Transcations on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.
7. Tin K. Ho. A Data Complexity Analysis of Comparative Adavantages of Decision Forest Constructors. *Pattern Analysis and Applications*, 5:102–112, 2002.
8. Tin K. Ho and M. Basu. Complexity Measures of Supervised Classification Problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):289–300, March, 2002.
9. S. Murthy, S. Kasif, and S. Salzberg. A System for Induction of Oblique Decision Trees. *Journal of Artificial Intelligence Research*, 2(1):1–32, 1994.
10. F.W. Smith. Pattern classifier design by linear programming. *IEEE Transactions on Computers*, C-17:367–372, 1968.
11. Stewart W. Wilson. Classifier Fitness Based on Accuracy. *Evolutionary Computation*, 3(2):149–175, 1995.
12. Stewart W. Wilson. Generalization in the XCS Classifier System. In J.R. Koza, W. Banzhaf, K. Chellapilla, K. Deb, M. Dorigo, D. Fogel, M.H. Garzon, D.E. Goldberg, H. Iba, and R. Riolo, editors, *Genetic Programming: Proceedings of the Third Annual Conference*. San Francisco, CA: Morgan Kaufmann, 1998.