On Classifier Domains of Competence

Ester Bernadó Mansilla
Computer Engineering Department
Ramon Llull University
esterb@salleURL.edu

Tin Kam Ho
Computing Sciences Research Center
Bell Labs, Lucent Technologies
tkh@research.bell-labs.com

Abstract

We study the domain of dominant competence of six popular classifiers in a space of data complexity measurements. We observe that the simplest classifiers, nearest neighbor and linear classifier, have extreme behavior of being the best for the easiest and the most difficult problems respectively, while the sophisticated ensemble classifiers tend to be robust for wider types of problems and are largely equivalent in performance. We characterize such behavior in detail using the data complexity metrics, and discuss how such a study can be matured for providing practical guidelines in classifier selection.

1. Introduction

Research and applications of pattern recognition suffer from a long-existing uncertainty concerning the optimal match between a method and a problem due to a strong dependence of classifier performance on data. This uncertainty is rooted in a lack of understanding on how data distributions interact with classifier geometry and the sampling processes. We believe that the key to improve upon the current level of automation in pattern learning is a better understanding of data set complexity in high-dimensional spaces, especially, the geometry of data distributions and its detailed relationship to classifier behavior.

In [4] a methodology is described where classification problems are characterized by a set of measures of the complexity of the class boundaries. The measures span a rich measurement space where one can compare the difficulty of different problems or different formulations of the same problem using alternative class definitions and feature transformations. They give quantitative descriptions on how close a problem is to being easily solvable (linearly separable) or to being intrinsically insolvable (as in a random labeling). We find that a collection of classification problems arising from real-world applications can span a large range in the values of these measures, and that they form a continuous and multi-facet distribution that has never been systematically studied. These problems present different degrees of difficulty to different kinds of classifiers. An attempt was made [3] to compare two complementary algorithms for decision forest construction to find out for what type of problems each method is preferable.

In this paper we extend this study further to compare a set of six popular classifiers. Our goal is to find out the domain of competence of each classifier, and to see if there exists any common behavior among these competing algorithms that are based on very different principles. We attempt to develop guidelines for recommending to practitioners the best suited classifier for each problem, using only measurable characteristics from the data. We report our unexpected finding of extreme behavior of two well known classifiers (linear classifier and nearest neighbor), and discuss our ideas on maturing this methodology for routine use.

2. Measures of Classification Complexity

We selected 9 metrics from [4] that describe the most important aspects of boundary complexity of a two-class problem (Table 1). These include classical measures for a feature's discriminating power as well as estimates of boundary lengths and class shapes via more sophisticated methods. Details of computation procedures are given in [4]. The 9 metrics span a measurement space where each problem, defined by a labeled data set, is represented by a point in this space. All measures are normalized as far as possible for comparability across problems.

`Boundary`	fraction of points on boundary estimated by MST
`Pretop`	fraction of points with maximal in-class ball retained
`IntraInter`	ratio of average intra/inter class NN distance
`NonLinNN`	nonlinearity of 1-nearest-neighbor(NN) classifi er
`NonLinLP`	nonlinearity of linear classifi er by linear programming
`Fisher`	maximum Fisher's discriminant ratio
`MaxEff`	maximum individual feature effi ciency
`VolumeOverlap`	volume of overlap region of class bounding boxes
`Npts/Ndim`	average number of points per dimension

Table 1. Complexity measures in this study.

3. Classifiers for Evaluation

We have chosen to evaluate 6 popular classifiers:

1. (`nn`) 1-Nearest neighbor by Euclidean distance.
2. (`lp`) Linear classifier constructed by linear programming minimizing sum of error distances [6].
3. (`odt`) Decision tree using oblique hyperplanes [5].
4. (`pdfc`) Random subspace decision forest [2].
5. (`bdfc`) Subsample decision forest, also known as *bagging*, or *bagged decision trees* [1].
6. (`xcs`) XCS, a genetic-algorithm based classifier using hyper-rectangular codification [7].

Among these classifiers, `nn`, `lp`, and `odt` are popular and standard classifiers in routine use. `pdfc`, `bdfc`, and `xcs` are newer classifiers developed in the last decade. They all take an ensemble learning approach, and are known to be highly robust and accurate for many practical problems. They are used in this evaluation because of the authors' familiarity with their implementation. We acknowledge that there are other popular and interesting classifiers to study once this methodology becomes mature, such as neural networks, support vector machines, boosting ensembles, and stochastic discrimination.

Many of the newer classifiers appear to be in close rivalry in benchmarking studies. This fact has created some overhanging questions: do they represent the limit of classification technology? what exactly have they added to the body of classifier methods? is there still value in the older methods? when exactly is each of them preferable? This study attempts to answer these questions.

4. Analysis Procedure

We evaluate the classifiers using 392 two-class problems from 14 UC-Irvine data sets (`abalone, car, german, kr-vs-kp, letter, lrs, nursery, pima, segmentation, splice, tic-tac-toe, vehicle, wdbc, and yeast`). We use pairwise class discriminations in these data sets that are shown to be linearly nonseparable. In the complexity space they span a large, multi-dimensional continuum ranging from nearly linearly separable to almost like random labeling [4]. The rich variations in their complexity provide an interesting domain for characterizing classifier performances. The complexity measures are computed for each problem using all available data points. There are no data source models for any of the problems, therefore we restrain our claims to be for the *apparent* complexity of the underlying problems as manifested in the given data sets.

We look for regions in the 9-dimensional complexity space where each classifier dominates, i.e., is significantly better than the others, and regions where multiple methods score similarly. Detailed steps are as follows.

1. For each problem and each method, we estimate the error rate by a 10-pass, two-fold cross validation. i.e., we split each data set randomly into two halves A & B, training a classifier in A and testing it on B. Then we train the classifier on B and test it on A. The counts of errors on both A & B are summed and divided by the data set size. The procedure is repeated for 10 passes using a new, random partition in each pass, resulting in 10 error rate estimates. The same 10 random partitions are used to evaluate each classifier.
2. For each data set, we consider the classifier with the lowest mean error rate (mean of the 10 estimates) to be the best method. Then we compare all other classifiers to the best using a paired t-test with a 95% confidence interval. We distinguish between those problems where one method *dominates*, i.e., the best method is significantly better than the others, from those where such difference is not significant, i.e., there are more than one methods statistically equivalent to the best.
3. The same procedure is used to find the worst method for each problem, and determine if its inferiority is significant.

We caution that the dominance conclusion is particular to the current pool of classifiers, in the sense that it could be changed by new classifiers added in a future study.

5. Observations

There are 270 problems (69% of all) that have a significantly best classifier and 157 problems (40%) that have a significantly worst classifier (Figure 1). The two sets overlap because some problems have both a significantly best and a significantly worst method. The remaining problems have several statistically equivalent best or worst methods.

Outstanding classifiers. Out of the six methods, there are four that are dominantly best for some problem: the nearest neighbor, the linear classifier, subsample forest (bagging), and XCS. The other classifiers (single decision tree and subspace forest) have lowest mean error rates for some problems, but their superiority is not dominant (statistically significant).

A remarkable discovery here is that almost all problems with a dominantly best classifier are best solved by either the nearest neighbor or the linear classifier (Table 2). These two simplest methods have *extreme* behavior. They are dominatingly best in many problems but can also be the worst in other problems. That means that they are very specialized methods. When the conditions are favorable for them, they perform optimally. Thus a challenge is to look for these conditions and determine whether they occur for a given problem.

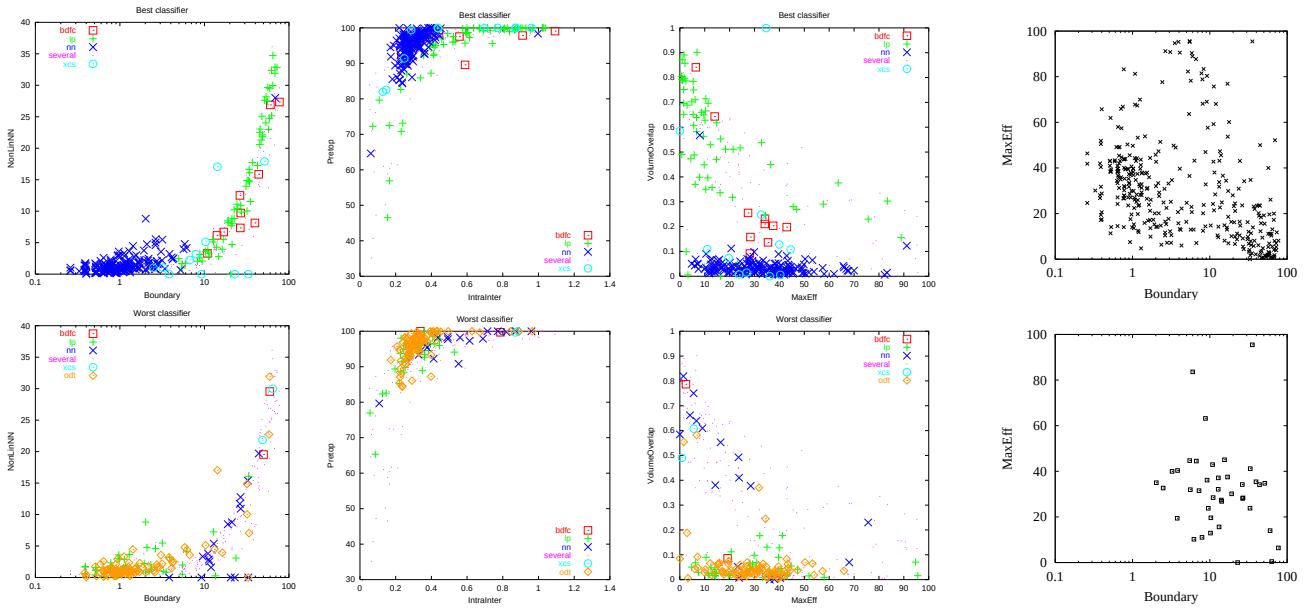

Figure 1. (Left) (top row) Problems with a dominant classifier (symbol) or several equivalent classifiers (.), shown in 3 projections of the complexity space; (bottom row) problems with a significantly worst classifier (symbol) or several bad classifiers (.), shown in the same projections. (Right) Problems where (top) `nn`, `lp`, or `odt` is best or equiv. to the best, (bottom) ensemble classifiers are needed.

For the problems with a significantly worst method, that method is most likely the single decision tree, followed by `lp` and `nn`. This reconfirms prior observations on the subsumption of a single decision tree by decision forests [3].

The ensemble classifiers, i.e., decision forests of both types and XCS, tend to be average performers. They behave similarly as a group, and they are rarely dominantly best or worst classifiers.

classifier	Best	%Best	Worst	%Worst
`nn`	186	69%	18	11%
`lp`	63	23%	47	30%
`odt`	0	0%	88	56%
`pdfc`	0	0%	0	0%
`bdfc`	10	4%	2	1%
`xcs`	11	4%	2	1%
total	270	100%	157	100%

Table 2. Distribution of dominating classifiers among problems with a significantly best or worst classifier.

Best suited domains. Apparently, from Figure 1, the domain of competence of `nn` locates in short boundaries and low nonlinearities. This means that `nn` is good for problems with compact, non-interleaving classes, or more specifically, with less than 10% of points on boundary, ratio of intra-inter class nearest neighbor distances less than 0.5 and NN-nonlinearities less than 6%. Outside this region, `nn` is hardly recommended.

Finding the domain of competence of the linear classifier is more difficult. The `lp` behavior is almost contrary to that of `nn`. It seems that for very short boundaries it performs worst, while it is best for most problems with boundary values between 10% and 70%. Although, for a few problems with boundaries inside this range (10%-70%) `lp` performs worst or as average. Analyzing the `lp` error w.r.t. other metrics, one is tempted to conclude that `lp` performs best when the problems are more difficult (long boundaries, high nonlinearities, large overlap volumes). For easy problems (short boundaries, low nonlinearities, etc.), `lp` is the worst method even though it has a low error rate. `lp` becoming the best method for difficult problems could be a symptom of that sparse training sets in those problems cause other classifiers to overfit [8].

The single decision tree is practically always outperformed by other classifiers so it has almost no domain of dominant competence. The ensemble classifiers are “average” methods for a wide range of problems, suggesting that when there is not enough information to apply a specialized classifier with strong confidence, they can give reasonable results. In these cases, XCS seems to perform better when

the classes are more compact, which is true also for subspace decision forest [3]. Bagging works better for longer boundaries and higher nonlinearities.

While there appears to be good separation between problems where `nn` or `lp` dominates, there is no clear signature that identifies problems where several methods are in close rivalry. One reason is that `nn` and `lp` are often among the several competing methods, and it is only when the other methods run into difficulties `nn` or `lp` becomes dominant. This could be caused by some ill conditions leading to ensemble overfitting that are not yet characterized by our metrics. Further studies are needed to identify such conditions.

Worst suited domains. The single decision tree and `lp` share almost the same worst suited domain, i.e., short boundaries, little class interleaving and small volume of class overlap. These classifiers should be avoided if such conditions occur. The nearest neighbor tends to be worst for boundary values greater than 10% and nonzero NN-nonlinearities. The worst domains for the ensemble classifiers are not obvious.

6. Discussions

Best metrics for classifier comparison. A by-product of the study is the identification of the metrics which are more relevant for discriminating between classifier domains of competence. These are: percentage of points on boundary, the nonlinearities, and the ratio of intra-inter class NN distances. These metrics are about the class geometry and the shape and the length of the class boundaries. Those describing the discriminative power of individual features, like `Fisher`, `MaxEff`, and `VolumeOverlap`, are less important for evaluating classifiers.

Some metrics can explain the behavior of some but not all classifiers. One reason for their narrower applicability is that their values are not spread uniformly with the current set of problems. For instance, `Pretop` has high values (over 80%) for almost all our problems. Although low values may indicate that the problem has a less complex geometry, we have too few such problems to extract useful conclusions.

Uneven distribution of sample problems. Our calculated fractions of problems where specific classifiers dominate are heavily dependent on the composition of the problem collection, and are not expected to be projectable to other collections. Moreover, there are empty regions in the complexity space where we do not know how the classifiers perform. We still do not know if these empty regions are induced by some geometrical constraints or are due to our particular choices of classification problems. For instance, in the sample problems, long boundaries occur often with high nonlinearities and intra-inter class distances. Although this correlation is reasonable, it is not necessarily true. Finally, problems from this particular archive have relatively small sizes and low dimensionalities, which may cause bias in the results. Using problems designed artificially to better cover the complexity space may overcome these difficulties.

Limitation of apparent complexity. Another source of difficulty that limits the strength of the conclusions is the estimation of complexity from given, fixed data sets without knowing how well they represent the underlying problems. This uncertainty needs to be quantified by statistical means.

7. Conclusions

We describe a methodology to compare a set of classifiers and to find their domains of competence. We find that the simplest classifiers, nearest neighbors and linear classifiers, have extreme behavior. They perform dominantly best when the conditions are suitable for them. The ensemble based classifiers are more robust, performing well for a wide range of problems with little differences among themselves. The linear classifier turns out to be the best for the most difficult kinds of problems, where failures of more sophisticated methods could be due to sample sparsity and overfitting.

Our study shows that the simplest classifiers still have good value when conditions are favorable, and that when the conditions are unknown and uncertain, the ensemble classifiers are reasonable choices. Yet our study is limited by the lack of uniformity in problem distributions and the lack of uncertainty characterization for the complexity estimates. We believe that this methodology can be much enhanced with theoretical studies on the influence of geometrical and topological constraints on problem distributions, better statistical procedures to quantify the uncertainties, and empirical studies adding in more diverse set of real-world problems and synthetic data sets.

Acknowledgements

Ester thanks the support of *Ministerio de Cienceia y Tecnologia* of Spain via project "TIC2002-040036-C05-03".

References

[1] L. Breiman, Bagging predictors, *Machine Learning*, **24**, 1996, 123-140.

[2] T.K. Ho, The random subspace method for constructing decision forests, *IEEE Trans. on PAMI*, **20**, 8, August 1998, 832-844.

[3] T.K. Ho, A data complexity analysis of comparative advantages of decision forest constructors, *Pattern Anal. & Appl.*, **5**, 2002, 102-112.

[4] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, *IEEE Trans. on PAMI*, **24**, 3, March 2002, 289-300.

[5] S. Murthy, S. Kasif, S. Salzberg, A System for Induction of Oblique Decision Trees, *J. of A.I. Research*, **2**, 1, 1994, 1-32.

[6] F.W. Smith, Pattern classifier design by linear programming, *IEEE Trans. on Computers*, **C-17**, 4, April 1968, 367-372.

[7] S.W. Wilson, Classifier Fitness Based on Accuracy, *Evolutionary Computation*, **3**, 2, 1995, 149-175.

[8] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.