A Characterization Of Difficult Problems In Classification

Ricardo Vilalta
IBM T.J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY. 10532 U.S.A.

Youssef Drissi
IBM T.J. Watson Research Center
19 Skyline Dr.
Hawthorne, NY. 10532 U.S.A.

Abstract ***Most classification algorithms experience difficulties when the input-output distribution is irregular (i.e., neighbor examples in the input space have different values in the output space), and sparse. We characterize the difficulty of a classification problem using a limited number of examples by re-defining a measure of class variation to account for irregular distributions, and by introducing a new measure of example cohesiveness in the input space to account for (lack of) sparseness. Our empirical study indicates these measures correlate with accuracy performance.***

keywords: classification, concept difficulty.

1 Introduction

The successful application of a classification algorithm is contingent on two factors: 1) a similarity between the hypothesis space and the class of concepts from which the target concept is derived, and 2) a sufficient number of training examples clustered in regions that are class-uniform. Since universal learning is unattainable [9], it is important to understand when a problem is difficult, either because the classification algorithm is not designed for certain kinds of input-output distribution, or because the examples available provide insufficient evidence to make a guess on that kind of distribution.

This paper is an attempt to characterize problems that are *difficult* to learn for most current classification algorithms. A common characteristic in most classification algorithms is the assumption that the distribution of examples in the input-output space is relatively smooth. This similarity-based bias [4][5] assumes examples lying close to each other in the input space share similar values in the output space (i.e., similar class value). A violation of this assumption complicates the process of delineating regions that are class-uniform.

The degree of lack of smoothness of the example distribution over the input-output space has been studied before, through a measure known as class variation [2][4]. Nevertheless, previous studies rely on complete knowledge of the input-output distribution. Our study extends previous work by re-defining class variation using a limited number of examples. In addition, we introduce a measure of the degree of cohesiveness of examples in the input space. Class variation and example cohesiveness are used together to understand learning difficulty: the former roughly measures the inadequacy of a similarity-based bias for the target concept, while the latter measures how dense or sparse is the example distribution.

Other measures of difficulty have been defined before. The VC dimension [1] characterizes the expressiveness of classes of concepts, in contrast to our approach that is focused on individual classification problems. The number of disjuncts in a DNF representation, entropy, and blurring (average entropy conditioned on each feature) are also different forms of assessing the degree of structure in the class distri-

bution [2][6]. Our approach is novel in that it weights the contribution to the amount of difficulty (class variation) based on the distance between examples, and it introduces a measure of example cohesiveness not previously linked to the difficulty of a classification problem. Experimental results show a correlation between our measures and predictive accuracy. Our analysis ends with the open problem of how to characterize individual misclassified examples.

The organization of this paper follows. Section 2 provides background information on classification and concept variation. Section 3 presents our characterization of difficulty in classification problems. Section 4 shows empirical results validating our measures. Section 5 ends with our conclusions.

2 Preliminaries

We define a classification problem P, as a 3-tuple $P =< T_{train}, m, \Phi >$, comprising a finite training set T_{train} of cardinality m, and a fixed but unknown probability distribution Φ. $T_{\text{train}} : \{(X_i, y_i)\}_{i=1}^{m}$ will consist of independently and identically distributed (i.i.d.) examples obtained according to Φ. Each example in T_{train} is an input-output pair. The first element of the pair, X_i, is a vector in an n-dimensional input space, $X_i = (x_i^1, x_i^2, \cdots, x_i^n)$. The second element, y_i, is the output value or class assigned to X_i by an unknown target function F, $F(X_i) = y_i$ (we assume a deterministic target function, i.e., zero-Bayes risk). The goal of classification is to find a hypothesis h that best approximates F, namely that minimizes a loss function (e.g., zero-one loss) in the input-output space, $\mathcal{X} \times \mathcal{Y}$, according to distribution Φ.

2.1 Class Variation

The degree of irregularity or difficulty of a concept F has been estimated before through a measure known as class variation ∇ [2][4]. ∇ estimates the probability that any two neighbor examples differ in class value, roughly measuring the amount of class irregularity. A highly irregular space is characterized by many small disjuncts, often requiring long concept representations. In contrast, a uniform space comprises large regions of examples sharing similar class labels, potentially allowing for compact concept descriptions.

The definition of ∇ works only in boolean spaces, and assumes all possible examples available. Let $X_1, X_2, \cdots, X_n$ be the n closest neighbors – at Hamming distance one – of an example X_i in an n-dimensional boolean space. Let the amount of concept variation for example X_i, $\sigma(X_i)$, be defined as

$$\sigma(X_i) = \frac{1}{n} \cdot \sum_{j=1}^{n} [1 - \delta(C(X_i), C(X_j))] \quad (1)$$

where $\delta(v, w) = 1$ if $v = w$ and 0 otherwise. Concept variation is defined as the average of this factor when applied to every example in the input space:

$$\nabla = \frac{1}{2^n} \cdot \sum_{i=1}^{2^n} \sigma(X_i) \quad (2)$$

The applicability of ∇ is limited to artificial concepts where all possible examples can be generated. It has been shown how concepts with high ∇ are difficult to learn by most conventional classification algorithms [2][4].

3 A Characterization Of Difficulty In Classification

This section gives a characterization of classification problems based on a more general definition of class variation. ∇ will be extended by considering only a limited sample of examples[1].

3.1 A Distance Metric

The definition of ∇ (Section 2.1) assumes we have access to the closest neighbors of a given example. Since in real-world domains that assumption no longer holds, a distance metric will be used to determine the contribution of

[1] A preliminary analysis is reported in [7].

each example to the amount of concept variation. Formally, let $X_i = (x_i^1, x_i^2, \cdots, x_i^n)$ and $X_j = (x_j^1, x_j^2, \cdots, x_j^n)$ be any two examples in training set T_{train}. We adopt the Euclidean distance between X_i and X_j as our distance metric:

$$D(X_i, X_j) = \sqrt{\sum_{k=1}^{n} d(x_i^k, x_j^k)^2} \tag{3}$$

where $d(x_i^k, x_j^k)$ is defined as follows for nominal and numeric features respectively:

$$d(x_i^k, x_j^k) = \begin{cases} 1 & \text{if } x_i^k \neq x_j^k \\ 0 & \text{if } x_i^k = x_j^k \end{cases} \tag{4}$$

$$d(x_i^k, x_j^k) = \frac{|x_k^i - x_k^j|}{\mathrm{NF}(x_k^i, x_k^j)} \tag{5}$$

NF is a normalization factor, e.g., $\mathrm{MAX}(X_k) - \mathrm{MIN}(X_k)$ (difference between the maximum and minimum values observed for feature k in T_{train}).

3.2 An Improved Measure Of Class Variation

Let us start by redefining class variation for a fixed example X_i. Our approach builds a (hyper) sphere around X_i of radius $r = 1$, and computes class variation inside the sphere as in equation 1 (n is now the number of examples inside the sphere). Radius r is computed according the the distance metric defined before (Section 3.1). Next we build a shell around the first sphere with an upper radius $r = 2$, and compute class variation on the examples lying within the shell (excluding examples within the sphere of $r = 1$). The process continues until all examples in the training set have been covered. In general, let $\tau(X_i, r)$ be the amount of class variation corresponding to the shell of upper radius r; class variation for X_i is defined as a weighted average over all $\tau(X_i, r)$:

$$\sigma(X_i) = \sum_r W(r) \cdot \tau(X_i, r) \tag{6}$$

The contribution to the amount of class variation for each shell will depend on the size of r; the smaller r, the closer the examples to X_i, and the higher the contribution. The following function will be used as the weight factor:

$$W(r) = \frac{1}{2^{\alpha \cdot r}} \tag{7}$$

where α is user defined and modulates the effect of distance (we use a default value of $\alpha = 1$). Now, let $m = |T_{\text{train}}|$ be the size of the training set. The new measure for concept variation, $\Upsilon(T_{\text{train}})$, is defined as the average variation over all training examples:

$$\Upsilon(T_{\text{train}}) = \frac{1}{m} \cdot \sum_{i=1}^{m} \sigma(X_i) \tag{8}$$

3.3 Example Cohesiveness

While Υ estimates the variability of class labels among examples, it fails to capture how dense or sparse is the example distribution in the training set. Our definition of example cohesiveness is similar to the weighted average used for class variation but attending to the number of examples exclusively. For a shell of upper radius r centered on example X_i, the density of the shell, $\phi(X_i, r)$, is defined simply as the number of examples lying inside the shell. Similarly as before, we define the cohesiveness of X_i, $\psi(X_i)$, as a weighted average over all $\phi(X_i, r)$:

$$\psi(X_i) = \sum_r W(r) \cdot \phi(X_i, r) \tag{9}$$

The new measure for example cohesiveness, $\Psi(T_{\text{train}})$, is defined as the average cohesiveness over all training examples:

$$\Psi(T_{\text{train}}) = \frac{1}{m} \cdot \sum_{i=1}^{m} \psi(X_i) \tag{10}$$

4 Variation Υ, Cohesiveness Ψ, and Learning Performance

We now show why Υ and Ψ are both important factors affecting the accuracy performance of a classification algorithm that adopts

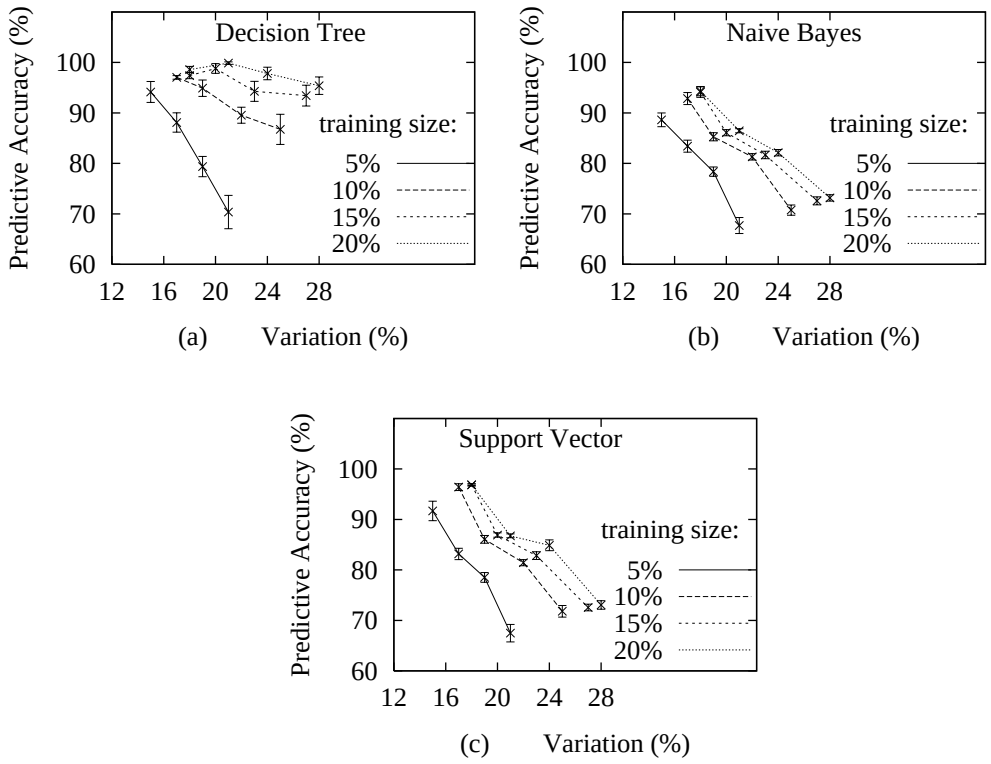

Figure 1: (a)-(c) Accuracy vs variation Υ using a decision-tree, a naive bayes, and a support vector machine.

a similarity-based bias (Section 1). We claim Υ intuitively captures the degree of irregularity of the input-output distribution around every example, while Ψ increases or decreases the confidence on Υ. Thus high values for Υ are offset by high values of Ψ, because a highly irregular distribution can be well approximated if we can place high confidence on our estimation.

As an illustration, consider the case of an example X_i and its immediate (uniformly distributed) neighborhood, defined by the examples within a sphere of radius $r = 1$ centered on X_i. If the degree of example cohesiveness around X_i, $\phi(X_i, 1)$, is maximum (e.g. $\phi(X_i, 1) = n$ in an n-dimensional boolean space), then a majority voting on the neighborhood of X_i will correctly guess the class of X_i (assuming a similarity-based bias is correct). But as the degree of cohesiveness, $\phi(X_i, 1)$, decreases (i.e., the distribution becomes sparse), the uncertainty in the class of X_i augments the probability of a misclassification. If $\phi(X_i, 1) = n - r$ and class variation is estimated as $\tau(X_i, 1)$, then the true class variation can take any value in the range:

$$[\frac{\tau(X_i,1)\cdot\phi(X_i,1)}{n}, \frac{\tau(X_i,1)\cdot\phi(X_i,1)+r}{n}] \tag{11}$$

Thus, in a worst-case scenario, the true class variation of X_i corresponding to its immediate neighborhood is offset by

$$\frac{\tau(X_i,1)\cdot\phi(X_i,1)+r}{n} - \tau(X_i,1) \tag{12}$$

Variation and cohesiveness are then inherently related to the difficulty of a classification problem.

4.1 Empirical Results

We now provide empirical support to our claims. To avoid the effects of noise, our

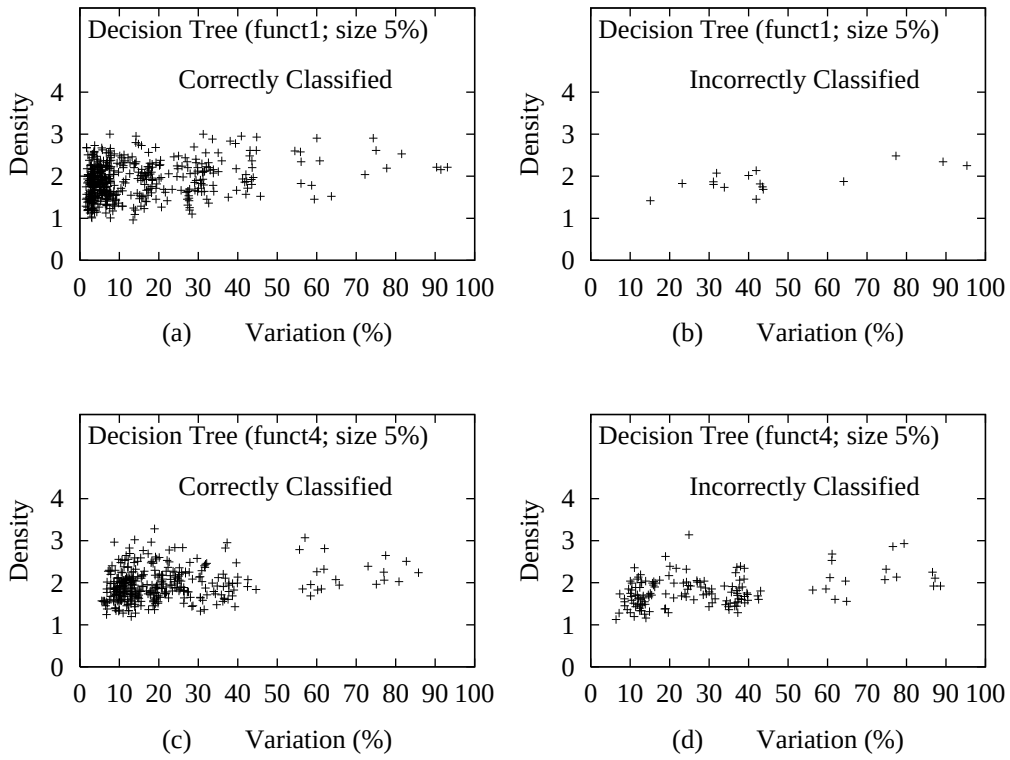

Figure 2: Cohesiveness ψ vs variation σ for correctly and incorrectly classified examples respectively on function 1 (a)-(b) and function 4 (c)-(d), using a decision tree.

experiments use four artificial boolean functions of increasing variation (defined in Appendix A). We use three classification algorithms: a decision-tree learner, a naive bayes, and a support vector machine. We use our own implementation for the decision tree algorithm that stops growing the tree if the number of examples on a node is less than $\beta = 3$ or if all examples are class uniform. The final tree is pruned using a pessimistic-pruning method [3]. The naive bayes and support vector machine implementations are part of the WEKA machine-learning class library [8], set using the default values. Results show means over 30 runs. Predictive accuracy is measured on the testing set exclusively. Confidence intervals are set to the 95% level assuming a two-sided normal distribution.

Our first experiments compare different values of variation vs predictive accuracy using all three classification algorithms (Figure 1 a-c). Each line corresponds to a fixed training-set size in the set $\{5\%, 10\%, 15\%, 20\%\}$, equivalent to different degrees of (fixed) cohesiveness. On every line, each point corresponds to a different boolean function (Appendix A). Results support our expectations of a decrease in predictive accuracy with increased variation[2]. As expected too, the results improve naturally with larger degrees of cohesiveness.

Our second set of experiments attempt to characterize individual misclassified examples, which we expect to group in areas of low cohesiveness and high variation. Figure 2 shows our results using the decision-tree algorithm. We fix the training-set size at 5%, where the highest difference in accuracy is observed between the functions with lowest and highest variation (function 1 and function 4 respectively). We analyze how the testing examples distribute

[2] The DNF nature of the artificial functions seems to favor the decision-tree algorithm. We ignore this fact and simply focus on the degradation of performance with increased variation.

based on their individual values of variation σ and cohesiveness ψ. Figure 2 (a)-(b) shows the distribution of correctly and incorrectly classified examples for function 1, while Figure 2 (c)-(d) shows the corresponding distribution for function 4. Surprisingly, incorrectly classified examples do not cluster in areas of high σ and low ψ. Function 4 has evidently higher global variation Υ because the center of (example) mass is to the right of the corresponding center of mass of function 1 (global cohesiveness Ψ is fixed for both functions since the training-set size is constant). Our proposed measures then give insight into the characterization of difficult classification problems but appear unable to characterize misclassified examples.

5 Conclusions

Our study shows that class variation Υ and example cohesiveness Ψ are critical factors in estimating the difficulty of a classification problem when a similarity-based bias holds. We show how high Υ degrades accuracy performance, but for fixed Υ, an increase in Ψ results in an increase of accuracy performance.

Our measures appear unable to make a distinction between correctly and incorrectly classified examples. This may stem from the local nearest-neighbor nature of our approach, in contrast to other approaches that may consider distributions of regions of examples. Thus, although the presence of dense clusters of uniform-class examples is favored by most classification algorithms, a characterization of misclassified examples seems highly dependent on the particular bias embedded by each classification algorithm. Future work will refine our measures to consider the particular bias embedded by standard classification algorithms.

Appendix A

Let: $X = (x_1, x_2, \ldots, x_9)$

Definitions:

function1 : $x_1x_2x_3 + x_1x_2x_4 + x_1x_2x_5$

function2 : $\bar{x}_1\bar{x}_2\bar{x}_3 + \bar{x}_2x_4\bar{x}_3 + \bar{x}_3\bar{x}_4\bar{x}_1$

function3 : $\bar{x}_5\bar{x}_3x_6 + \bar{x}_6x_8\bar{x}_5 + x_8\bar{x}_3\bar{x}_2$

function4 : $\bar{x}_6x_1x_8 + x_8x_4\bar{x}_1 + \bar{x}_9\bar{x}_8x_1$

Acknowledgments

We are grateful to Mark Brodie for his valuable suggestions. This work was supported by IBM T.J. Watson Research Center (New York, USA).

References

[1] Blumer, A. & Ehrenfeucht, A. & Haussler, D. & Warmuth, K. (1989) Learnability and the Vapnik-Chervonenkis Dimension. *Journal of the Association for Computing Machinery*, **36**, pp. 929-965.

[2] Pérez, E. & Rendell, L. A. (1996) Learning Despite Concept Variation by Finding Structure in Attribute-Based Data. In L. Saitta (eds.), *Proceedings of the Thirteenth International Conference on Machine Learning*, pp. 447-455. Morgan Kaufmann.

[3] Quinlan, J. R. (1994) *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

[4] Rendell L. (1990) Learning Hard Concepts Through Constructive Induction: Framework and Rationale. *Computational Intelligence*, **6**, pp. 247-270.

[5] Rendell L. (1990) Empirical Learning as a Function of Concept Character. *Machine Learning*, **5**, pp. 267-298.

[6] Rendell L., & Ragavan H. (1993) Improving the Design of Induction Methods by Analyzing Algorithm Functionality and Data-Based Concept Complexity. *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 952-958.

[7] Vilalta R. (1999) Understanding Accuracy Performance Through Concept Characterization And Algorithm Analysis. *Workshop on Recent Advances in Meta-Learning and Future Work, 16th International Conference on Machine Learning*, pp. 3-9.

[8] Witten I. H. & Frank E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Academic Press, London U.K.
http://www.mkp.com/datamining.

[9] Wolpert, D. (1996) The Lack of a Priori Distinctions Between Learning Algorithms and the Existence of a Priori Distinctions Between Learning Algorithms. *Neural Computation*.