An Empirical Study of Instance Hardness

Michael R. Smith, Tony Martinez, Christophe Giraud-Carrier
Computer Science Department
Brigham Young University
Provo, UT 84602
msmith@axon.cs.byu.edu, {martinez,cgc}@cs.byu.edu

Abstract

Most performance metrics for learning algorithms do not provide information about the misclassified instances. Knowing which instances are misclassified and understanding why they are misclassified could guide future algorithm development. In this paper, we analyze the classification of over 190,000 instances from 64 data sets and create heuristics to analyze and predict an instance's expected difficulty to classify correctly (*instance hardness*). We find that 5% of the instances are misclassified by all 9 considered learning algorithms and that 17% are misclassified by at least half. The principal contributor to misclassification is class overlap. We demonstrate the utility of instance hardness by using it to filter hard instances from the data sets which increases the classification accuracy on test data (including the hard instances).

1 Introduction

Algorithmic development for classification problems has been measured by classification accuracy, precision, or a similar metric on benchmark data sets. These metrics, however, only provide aggregate information about the learning algorithm and the task upon which it operates. They do not provide information about which instances are misclassified and why they are misclassified. Understanding why instances are misclassified can shed light on the field of machine learning and could further lead to the development of learning algorithms that address the causes of misclassification.

Previous work has studied hard instances at the instance level from the premise that outliers [1], boundary points [2], or instances belonging to a minority class [3] are hard to classify. These experiments have generally been carefully designed around one of these issues. For example, outlier detection often uses artificial data sets or systematically injects noise into well-known data sets [4]. At the data set level, meta-learning has examined the complexity of the data sets and which learning algorithm to use. Ho and Basu [5] focused on the geometrical complexity of the data on 2-class problems. Prior work have also mostly considered only a limited number of data sets and algorithms.

To understand why instances are misclassified, we empirically analyze over 190000 instances from 57 UCI data sets and 7 non-UCI data sets classified by 9 learning algorithms. We focus explicitly on instances that are misclassified by most of the considered learning algorithms, and seek to shed light on the reasons for such misclassifications. The diversity of learning algorithms and unaltered data sets allows us to offer insight into why instances are misclassified independent of the learning algorithm and task. We also propose a set of heuristics to identify instances that can reasonably be expected to be misclassified and compare them to the instances that are misclassified by all or most of the learning algorithms[1]. The combination of the heuristics can be used to predict instance

[1] By "reasonably expected to be misclassified", we mean that in the absence of additional information beyond what the data set provides, the label assigned by the learning algorithm to the instance is the most appropriate one, even if it happens to be different from the instance's target value.

hardness on novel instances. We find that class overlap most directly affects classification accuracy. Class overlap refers to how similar an instance is to instances of a different class. Also, class skew alone does not affect classification accuracy but exacerbates the effects of class overlap.

As an application of instance hardness, we removed the instances with a high degree of class overlap from the data sets during training and observe an increase in classification accuracy on test data (including the removed hard instances) for all of the learning algorithms. The accuracy on the hard instances decreases (as expected), yet it increases sufficiently on the other instances to provide an increase in overall accuracy. Thus, the learning algorithms are less prone to overfit and define a classification boundary that is more representative of the data.

The remainder of the paper is organized as follows. Section 2 reviews previous work. Section 3 presents the methodology and heuristics. Sections 4 and 5 provide an analysis of hardness at the instance-level and at the data set-level respectively. Section 6 examines the impact on accuracy of removing hard instances for noise reduction. The paper concludes in Section 7.

2 Related Work

We are not the first to examine instances that are hard to classify correctly. Prior work has examined hard instances from the premise that they are outliers, border points, or belong to a minority class.

Outlier detection has received growing attention from the data mining community where outliers may represent anomalies or points of focus [6].There are many outlier detection algorithms from a variety of fields using different approaches. For example, Local Outlier Factor (LOF) [7] is an approach loosely related to density-based clustering that assigns each instance a value representing its potential of being an outlier with respect to the instances in its neighborhood. A thorough survey of outlier detection methodologies is provided by Hodge and Austin [8].

Most of the attention for border instances has come from instance reduction techniques to avoid storing more instances than are necessary to generalize well on the data [2]. Wilson and Martinez [9] present a survey of instance-based reduction techniques as well as propose their own. These and similar algorithms attempt to smooth the decision boundary by removing outliers and by only keeping enough boundary instances to maintain good classification accuracy. On the other hand, some instance-based reduction techniques only keep a central representation of the instances and discard the outliers and some border points [10].

Class skew refers to a data set consisting of one or more classes heavily outnumbering the other class(es) and has been observed to make instances harder to classify correctly [11]. Many learning algorithms have difficulties learning the concepts of the minority class(es). Most previous work has used undersampling, oversampling, and cost-sensitive techniques and has been limited to binary classification tasks. Class skew can also affect outliers and boundary instances. Akbani et al [12] use SMOTE [13] (an oversampling technique) in conjunction with SVMs to address the class imbalance problem. The resultant support vectors provide information about the class boundaries.

Our work also relates to meta-learning. Meta-learning uses data sets features to predict which learning algorithm to use and/or the learning algorithms performance on the data set [14, 15]. While in meta-learning the prediction is driven by accuracy and thus performance at the data set level, we focus on the instance level. Using heuristics in conjunction with the classification of various learning algorithms we characterize instances which are commonly misclassified rather than suggesting the proper learning algorithm to use, although future work could include this direction.

Previous work has focused on a single issue at a time whereas we do not focus on a single issue as a cause for an instance being misclassified. We focus our analysis on discovering the underlying causes for instances being misclassified from a broad perspective. Our analysis is extensive in the number of learning algorithms and the number of data sets. Also, we do not alter the data sets.

3 Experimental Methodology

We investigate instances that are hard to classify by analyzing the instances from 57 UCI data sets [17]. The data sets are classified using a collection of nine learning algorithms drawn from various model classes shown in Table 1. The learning algorithms are used as implemented in Weka with their

Table 1: List of learning algorithms.

Learning Algorithms	
Decision Tree (C4.5 [16])	Naïve Bayes
Multi-layer Perceptron trained with Back Propagation	Perceptron
Support Vector Machine	1-NN (1-nearest neighbor)
5-NN (5-nearest neighbors)	RIPPER
Radial Basis Function Network	

default parameters [18]. By adjusting the parameters, some instances may be correctly classified more consistently. However, parameter optimization is an expensive and non-trivial process, beyond the skills of most users. Hence, using default parameters gives insight into which instances are misclassified in most practical scenarios.

We emphasize the extensiveness of our analysis. We examine 178,109 instances individually. A total of 5130 models are produced from 9 learning algorithms trained with 57 data sets using 10-fold cross-validation. With this volume and diversity, our results provide useful information about the extent to which hard instances exist and what contributes to instance hardness.

We first identify which instances are misclassified. Next, we use a set of heuristics to analyze both the extent and the nature of misclassifications in typical machine learning tasks. We also examine 12,233 instances from a test set of seven non-UCI data sets not used to generate the hardness heuristics to ensure that the heuristics generalize well [19, 20, 21, 22].

To identify which instances are hard to classify we define *instance hardness* as the average number of learning algorithms which incorrectly classify an instance.

$$instance\ hardness(x) = \frac{\sum_{i=1}^{N} incorrect(LA_i, x)}{N}$$

where x is the data instance, N is the number of learning algorithms, and $incorrect(LA_i, x)$ returns 1 if an instance x is misclassified by learning algorithm LA_i, and 0 otherwise. The hardest instances are those which no learning algorithm correctly classifies. Their hardness value is 1. To obtain an aggregate value of hardness for a complete data set we define *data set hardness* by averaging instance hardness over the instances in a data set. The definition of hardness depends on the set of selected learning algorithms. This is an appropriate basis, however, as it focuses on instances that current machine learning approaches misclassify. As it is not possible to know an instance's actual hardness value, our definition provides a good approximation.

To characterize and analyze the instances that are hard to classify empirically designed a set of seven heuristics (*hardness heuristics*). These heuristics use the bias from various learning algorithms (similar to landmarking [23]) to analyze and identify instances that may be misclassified more frequently.

The first heuristic, *k-Disagreeing Neighbors* (*k*DN), measures the local overlap of an instance in the original task space. The *k*DN of an instance is the percentage of that instance's k nearest neighbors (using Euclidean distance) that do not share its target class value.

$$kDN(x) = \frac{\mid \{y : y \in \mathit{kNN}(x) \wedge t(y) \neq t(x)\} \mid}{k}$$

where $\mathit{kNN}(x)$ is the set of k nearest neighbors of x and $t(x)$ is the target class for x.

The next heuristic measures how tightly a learning algorithm has to divide the task space to correctly classify an instance and the complexity of the decision boundary. Some learning algorithms, such as decision trees and rule-based learning algorithms, can express the learned concept as a disjunctive description. Thus, the *Disjunct Size* (DS) of an instance is the number of instances in a disjunct divided by the number of instances covered by the largest disjunct in a data set.

$$DS(x) = \frac{\mid disjunct(x) \mid -1}{\max_{y \in D} \mid disjunct(y) \mid -1}$$

where the function $disjunct(x)$ returns the disjunct that covers instance x, and D is the data set that contains instance x. The disjuncts are formed using a C4.5 [16] decision tree, created without pruning and setting the minimum number of instances per leaf node to 1[2].

The third heuristic measures an instance's overlap on a subset of the features. Using a pruned C4.5 tree, the *Disjunct Class Percentage* (DCP) of an instance is the number of instances in a disjunct belonging to its class divided by the total number of instances in the disjunct.

$$DCP(x) = \frac{\mid \{z : z \in disjunct(x) \wedge t(z) = t(x)\} \mid}{\mid disjunct(x) \mid}$$

The fourth heuristic provides a global measure of overlap and the likelihood of an instance belonging to a class. The *Class Likelihood* (CL) of an instance belonging to a certain class is defined as

$$CL(x, t(x)) = \prod_{i}^{|x|} P(x_i|t(x))$$

where x_i is the value of instance x on its ith attribute[3]. The prior term is excluded in order to avoid bias against instances that belong to a minority classes.

The fifth heuristic captures the difference in likelihoods and global overlap. The *Class Likelihood Difference* (CLD) is the difference between the class likelihood of an instance and the maximum likelihood for all of the other classes.

$$CLD(x, t(x)) = CL(x, t(x)) - \underset{y \in Y - t(x)}{\operatorname{argmax}} CL(x, y)$$

The sixth heuristic captures the skewness of the class an instance belongs to. For each instance, its *Minority Value* (MV) is the ratio of the number of instances sharing its target class value to the number of instances in the majority class.

$$MV(x) = \frac{\mid \{z : z \in D \wedge t(z) = t(x)\} \mid}{\max_{y \in Y} \mid \{z : z \in D \wedge t(z) = y\} \mid}$$

The final heuristic offers an alternative to MV. If there is no class skew, then there is an equal number of instances for all classes. Hence, the *Class Balance* (CB) of an instance is:

$$CB(x) = \frac{\mid \{z : z \in D \wedge t(z) = t(x)\} \mid}{\mid D \mid} - \frac{1}{\mid Y \mid}.$$

If the data set is completely balanced the class balance value will be 0.

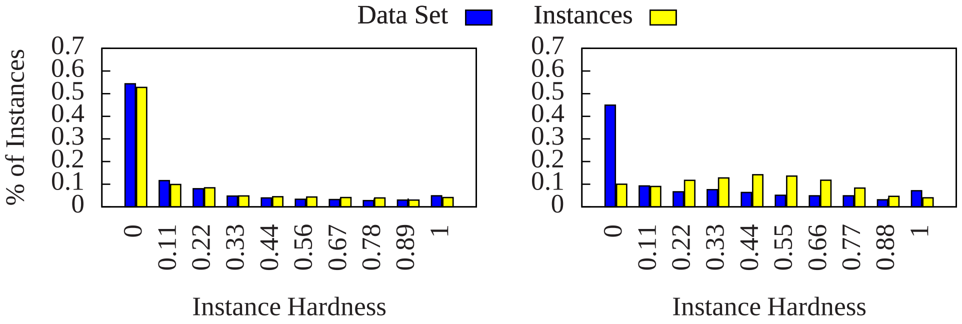
4 Instance-level Analysis

Figure 1 shows the percentage of instances per instance hardness value for the UCI and non-UCI data sets. Given 9 learning algorithms, there are 10 possible levels of instance hardness, ranging from 0 (classified correctly by all algorithms) to 1 (misclassified by all algorithms). The first column shows the percentage of instances averaged per data set and the second column shows the percentage over all instances. We use the values averaged over all data sets so as not to be biased towards larger data sets. Also, there are considerably more hard instances in the non-UCI data sets. This is due to the high number of UCI data sets that are easy to classify.

These results show that a significant amount of instances are hard: 5% of the instances from the UCI data sets are misclassified by all of the learning algorithms and 17% are misclassified by at least half. For the instances from the non-UCI data sets, 7% are misclassified by all of the learning algorithms and 25% are misclassified by at least half. Seeking to improve our understanding of why these instances are misclassified becomes a justifiable quest.

[2] Note that C4.5 will create fractional instances in a disjunct for instances with unknown attribute values, possibly leading to DS values less than 1. Such cases are treated as though the disjunct covered a single instance.

[3] Continuous variables are assigned a probability using a kernel density estimation [24].



a) UCI data sets

b) non-UCI data sets

Figure 1: Overall instance hardness

Table 2: The correlation coefficients for the hardness heuristics relating to instance hardness.

Heuristics:	DN	DS	DCP	CL	CLD	MV	CB
UCI	**0.8487**	0.4034	0.6757	0.7726	0.7342	0.4702	0.3138
non-UCI	**0.6363**	0.3829	0.2055	0.6101	0.5953	0.0954	0.2078

We consider the relationships between instance hardness and the hardness heuristics and present an additional set of heuristics for identifying instances as having high, low, or no overlap and belonging to a minority class. Table 2 provides the correlation coefficients from a linear regression model for each hardness heuristic on the UCI and non-UCI data sets. The data from the UCI data sets was used to generate the model. The heuristics that measure class overlap (DN, DCP, CL, and CLD) have significantly larger correlation coefficients than those that measure the decision boundary complexity and class skew (DS, MV and CB). We also examined the relationships of all the heuristics together. The instance hardness and heuristic values from the UCI data sets for each instance were compiled and linear regression was used to predict instance hardness. The resulting model is as follows.

$$\begin{aligned} instance\ hardness =& 0.5569 * DN - 0.1984 * DCP - 0.124 * CL + 0.0752 * CB \\ & -0.072 * CLD + 0.0365 * DS + 0.0339 * MV + 0.9088 \end{aligned}$$

with a correlation coefficient of 0.8856 on the UCI data sets using 10-fold cross-validation and 0.7302 on the non-UCI data sets. DN, DCP, and CL have the largest coefficients (only DN is statistically significant using the *t*-test with a *p* value of 0.05) suggesting that overlap is the most informative for predicting instance hardness. There is no heuristic for class skew in the equation, which coincides with Batista's conclusion that class skew alone does not hinder learning algorithm performance [25].

We observe that combining DCP and DS provides more information about instance hardness than they do individually. 99% of the instances with instance hardness value 1 and DS value 1 have a DCP value less than 0.5. Using these observations we identify instances with high, low, or no overlap using the following heuristic.

$high$ if $(CLD(x,t(x)) < 0$ &&$((DS(x) == 0$ && $DCP(x) < 0.5)$ || $DN(x) > 0.8))$
low else if $((DS(x) == 0$ && $DCP(x) < 1)$ || $DN(x) > 0.2)$
$none$ otherwise.

The high overlap instances are those that have a higher class likelihood for the wrong class and do not agree with 80% of their nearest neighbors[4] or the learning algorithm had to overfit the data to correctly classify it. An instance has low overlap if it does not have high overlap and it does not agree with at least 80% of its neighbors or the disjunct it belongs to is not pure. Otherwise, the instance is identified as having no overlap.

An instance is identified as belonging to a minority class if the number of instances in the class is less than or equal to half the number of instances belonging to the majority class ($MV(x) \leq 0.5$),

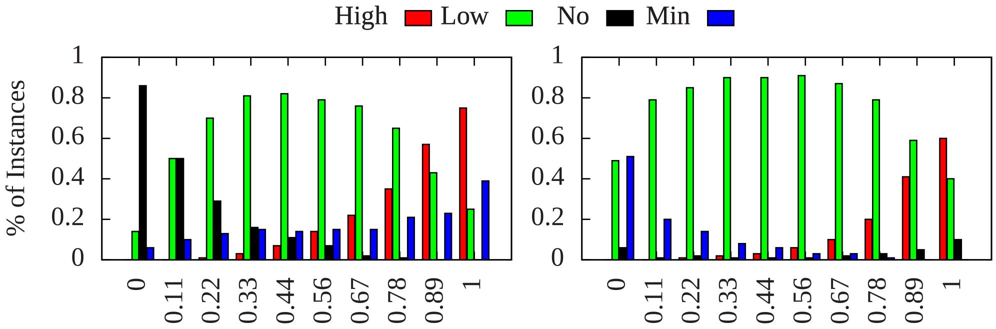
[4] To factor out the effect of neighborhood size, we use $DN(x)$ rather than $kDN(x)$, where $DN(x)$ is the average of $kDN(x)$ over all values of k between 1 and 17. Setting DN above 0.8 implies that on average, for every 5 instances in the neighborhood, at least 4 disagree with the instance under consideration.

Table 3: Percentage of instances that were misclassified according to instance type.

Instance Type	High	Low	None	Min	MinHigh	MinLow	MinNone
% Misclassified (UCI)	83.0	35.0	3.4	41.9	88.2	41.8	3.8
% Misclassified (non-UCI)	78.6	44.3	16.1	48.9	86.0	51.1	1.1

and the number of instances in the class is less than the number of instances if all classes were balanced ($CB(x) < 0$).

An analysis of the instances and their hardness heuristics shows that class overlap is a principal contributor to instance hardness. As instance hardness increases, there is an increase in high overlap instances and a decrease in no overlap instances. This is shown in Figure 2 which gives the percentage of instances with high overlap, low overlap, no overlap, and class skew according to instance hardness. The non-UCI data sets have considerably less no overlap instances and more low overlap instances giving insight into why the non-UCI data sets are more difficult to classify.



a) UCI data sets

b) non-UCI data sets

Figure 2: Instances with high, low, and no overlap and minority class according to instance hardness.

Table 3 gives the percentage of instances that were misclassified according to the amount of overlap and class skew. For the UCI and non-UCI data sets, about 80% of the instances with high overlap were misclassified whereas only about 40% of the low overlap instances were misclassified. Hence, class overlap is a contributing factor to misclassification. The percentage of the no overlap instances is significantly higher on the non-UCI data sets since the non-UCI data sets are generally more difficult than the UCI data sets.

Class skew alone does not cause misclassifications. However, of all the misclassified instances that belong to a minority class, about 65% also have high or low overlap. The percentage misclassified for the high and low overlap increases when the instance also belongs to a minority class (Min and High, Min and Low). This suggests that class overlap is exacerbated by class skew.

5 Data Set-level Analysis

We also examine hardness at the data set level using our heuristics. We compare against a set of complexity measures by Ho and Basu [5] (implemented with DCoL [26]) and a set of meta-learning features from Brazdil et al [27]. The complexity measures and meta-features are shown in Table 4.

We examined each heuristic and complexity measure individually to determine how well it predicts data set hardness. The measures that account for overlap are the best at indicating data set hardness. The average data set hardness for the data sets with the top 10 average DN values is 0.473 (the average for all data sets is 0.202). N1 was the most indicative of data set hardness from the set by Ho and Basu with an average data set hardness value of 0.423 for the 10 data sets with highest N1 value. From Brazdil's meta-features, the entropy of classes had the highest average data set hardness

Table 4: List of complexity measures.

Complexity	L2: Error rate of linear classifier by LP	L3: Nonlinearity of linear classifier by LP
	N1: Fraction of points on class boundary	N2: Ratio of ave intra/inter class NN dist
	N3: Error rate of 1NN classifier	N4: Nonlinearity of 1NN classifier
	T1: Fraction of maximum covering spheres	T2: Ave number of points per dimension
	F3: Max individual feature efficiency	
Meta	Number of instances	Number of attributes
	Proportion of nominal/real attributes	Proportion of attributes with outliers
	Entropy of classes	

values for the meta-features. The average data set hardness values for the heuristics that measure class skew are lower than the average of all the data sets and thus are not good indicators of data set hardness. In general the meta-features are not a good indicator of data sets hardness which is not surprising as their goal is to predict which learning algorithm to use.

Applying linear regression to estimate data set hardness based on data set features also shows what contributes to data set hardness. The result is as follows.

$$\begin{aligned} data\ set\ Hardness =& 0.4539 * DN - 0.4314 * CL - 0.2111 * DCP + 0.088 * CLD \\ & +0.0763 * N3 - 0.047 * N4 - 0.034 * F3 + 0.0239 * F4 + 0.4815 \end{aligned}$$

with a correlation coefficient of 0.9562 using 10-fold cross-validation on the UCI data sets and 0.7939 on the non-UCI data sets. Using just the complexity measures resulted in a correlation coefficient of 0.4361. The addition of the complexity measures slightly decreased the correlation coefficient of a linear regression model using just the hardness heuristics from 0.9586. This shows that the hardness heuristics are better suited for determining data set hardness than the complexity measures from Ho and Basu. The most highly weighted features, and the only features with coefficients that are statistically significant using the *t*-test with a *p* value of 0.05, are DN, and CL which further supports the claim that class overlap is a principal cause for instance and data set hardness. As with instance hardness, class skew is not significant in the linear regression equation.

6 Noise Reduction

In this section we briefly demonstrate an example application of instance hardness. The instances that have a high degree of class overlap are possibly mislabeled or noisy instances. Class noise reduction methods have shown that removing mislabeled and noisy instances for training increases the classification accuracy [28, 29, 30, 31]. Here, we remove the instances identified as having high overlap for training, but include them for evaluation. We identify high overlap instances using the hardness heuristics (NoHOL) and instances that have a predicted instance hardness value greater than or equal to a threshold value using the linear regression equation in Section 4. We use threshold values of 0.5 and 1 for the linear regression equation (LR 0.5 and LR 1). All 53 of the 57 UCI data sets that contain instances with high overlap and all of the non-UCI data sets are evaluated using 10-fold cross-validation. We compare the results with the Repeated Edited Nearest Neighbor (RENN) algorithm for noise reduction [28] and majority and consensus ensemble filters [29] using the nine learning algorithms in this study. Statistical significance is tested using the Wilcoxon signed-ranked test [32].

Filtering the instances for training increases the classification accuracy for all of the considered learning algorithms. Examining the increase in accuracy according to the percentage of high overlap instances present in the data sets shows that removing instances with high overlap is more beneficial for data sets that have more than 10% high overlap instances (HI). This is shown in Table 5 which gives the average accuracy for the nine considered learning algorithms on all instances and broken down according to the high, low, or no overlap. The benefit of using instance hardness is most clearly seen on the high overlap instances where their average accuracy is the lowest. This is desired because the instances with high overlap are likely noisy instances and should be misclassified based on the instance labels. The instance hardness methods are competitive with RENN and the ensemble methods despite decreasing on the high overlap instances. The changes in accuracy are statistically significant ($\alpha = 0.05$) for all cases with respect to the original dataset.

Table 5: Average accuracy on the filtered and original data sets.

		UCI			non		UCI			non
OL	Train	HI	Low	All	All	Train	Hi	Low	All	All
All	Original	0.571	0.847	0.785	0.723					
	NoHOL	0.599	0.856	0.798	0.745	RENN	0.598	0.859	0.800	0.753
	LR 1	**0.627**	0.849	0.799	0.733	Con	0.600	0.857	0.798	0.743
	LR 0.5	0.605	0.857	0.799	**0.759**	Maj	0.621	**0.864**	**0.809**	0.750
High	Original	0.121	0.174	0.162	0.130					
	NoHOL	**0.083**	**0.148**	**0.134**	0.117	RENN	0.124	**0.141**	0.137	0.099
	LR 1	0.120	0.167	0.156	0.132	Con	0.122	0.162	0.153	0.117
	LR 0.5	**0.087**	**0.141**	**0.129**	**0.072**	Maj	0.162	0.159	0.159	0.092
Low	Original	0.636	0.706	0.690	0.641					
	NoHOL	0.690	**0.723**	0.716	0.690	RENN	0.674	**0.720**	0.709	**0.714**
	LR 1	0.630	0.703	0.687	0.641	Con	0.678	**0.720**	0.711	0.661
	LR 0.5	0.680	0.713	0.706	0.704	Maj	**0.703**	**0.723**	**0.722**	**0.712**
None	Original	0.924	0.967	0.958	0.946					
	NoHOL	0.968	0.972	0.971	0.952	RENN	0.971	0.973	0.973	0.960
	LR 1	0.928	0.968	0.960	0.947	Con	**0.985**	0.971	0.974	0.954
	LR 0.5	0.980	**0.978**	**0.979**	**0.970**	Maj	**0.989**	**0.977**	**0.980**	0.961

7 Conclusion and Future Work

We empirically analyzed to what extent instances are hard to correctly classify. Our analysis was extensive, examining 64 data sets, over 190,000 instances, and 9 learning algorithms. We generated over 5200 models. We found that there is a set of instances that all learning algorithm misclassify.

We presented a set of hardness heuristics to identify instances that are hard to classify correctly. These heuristics indicate that class overlap most directly affects instance hardness. Class skew alone does not make an instance hard to classify correctly unless it is an issue of data underrepresentation. However, in the presence of class overlap, class skew exacerbates the difficulties of class overlap.

We showed that our heuristics can also be used to preprocess data sets by removing instances with high overlap for training. This improved classification accuracy for all of the considered learning algorithms, most notably on data sets with a high percentage of high overlap instances. By removing these instances, the learning algorithms could better determine the classification boundary and improve their classification accuracies.

Future work could include weighting the instances for training based on the hardness heuristics and developing learning algorithms designed to be more robust to overlap. By knowing which instances should be misclassified, new evaluation methods could be used to assess the performance of learning algorithms based on which instances where correctly classified as well as misclassified.

References

[1] N. Abe, B. Zadrozny, and J. Langford. Outlier detection by active learning. In *Proceedings of the 12th international conference on Knowledge discovery and data mining*, pages 504–509. ACM, 2006.

[2] H. Brighton and C. Mellish. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6(2):153–172, 2002.

[3] J. van Hulse, T. M. Khoshgoftaar, and A. Napolitano. Experimental perspectives on learning from imbalanced data. In *Proceedings of the 24th international conference on Machine learning*, pages 935–942, New York, NY, USA, 2007. ACM.

[4] T. M. Khoshgoftaar, N. Seliya, and K. Gao. Rule-based noise detection for software measurement data. In *Proceedings of the IEEE International Conference on Information Reuse and Integration*, pages 302–307, 2004.

[5] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24:289–300, March 2002.

[6] J. M. Kubica and A. Moore. Probabilistic noise identification and data cleaning. In *The Third IEEE International Conference on Data Mining*, pages 131–138. IEEE Computer Society, November 2003.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. *SIGMOD Record*, 29(2):93–104, June 2000.

[8] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[9] D. R. Wilson and T. R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.

[10] J. Zhang. Selecting typical instances in instance-based learning. In *Proceedings of the ninth international workshop on Machine learning*, pages 470–479, 1992.

[11] S. Hido and H. Kashima. Roughly balanced bagging for imbalanced data. In *Proceedings of the SIAM International Conference on Data Mining*, pages 143–152. SIAM, 2008.

[12] R. Akbani, S. Kwek, and N. Japkowicz. Applying support vector machines to imbalanced datasets. In *Proceedings of the 15th European Conference on Machine Learning (ECML)*, pages 39–50, 2004.

[13] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research (JAIR)*, 16:321–357, 2002.

[14] J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149–198, 2000.

[15] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40(3):203–228, 2000.

[16] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, USA, 1993.

[17] A. Asuncion and D. J. Newman. UCI machine learning repository, 2007.

[18] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Fransisco, 2nd edition, 2005.

[19] K. Thomson and R. J. McQueen. Machine learning applied to fourteen agricultural datasets. Technical Report 96/18, The University of Waikato, September 1996.

[20] J. Salojärvi, K. Puolamäki, J. Simola, L. Kovanen, I. Kojo, and S. Kaski. Inferring relevance from eye movements: Feature extraction. Technical Report A82, Helsinki University of Technology, March 2005.

[21] J. Sayyad Shirabad and T. Menzies. The PROMISE Repository of Software Engineering Databases. School of Information Technology and Engineering, University of Ottawa, Canada, 2005.

[22] G. Stiglic and P. Kokol. GEMLer: Gene expression machine learning repository. University of Maribor, Faculty of Health Sciences, 2009.

[23] B. Pfahringer, H. Bensusan, and C. G. Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, pages 743–750, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[24] G. H. John and P. Langley. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345, 1995.

[25] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations Newsletter*, 6(1):20–29, 2004.

[26] A. Orriols-Puig, N. Macià, E. Bernadó-Mansilla, and T. K. Ho. Documentation for the data complexity library in c++. Technical Report 2009001, La Salle - Universitat Ramon Llull, April 2009.

[27] P. B. Brazdil, C. Soares, and J. P. D. Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277, 2003.

[28] I. Tomek. An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:448–452, 1976.

[29] C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

[30] D. Gamberger, N. Lavrač, and S. Džeroski. Noise detection and elimination in data proprocessing: Experiments in medical domains. *Applied Artificial Intelligence*, 14(2):205–223, 2000.

[31] J. S. Sánchez, R. Barandela, A. I. Marqués, R. Alejo, and J. Badenas. Analysis of new techniques to obtain quality training sets. *Pattern Recognition Letters*, 24:1015–1022, April 2003.

[32] J. Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.