Addressing the Curse of Imbalanced Training Sets: One-Sided Selection

Miroslav Kubat and Stan Matwin
Department of Computer Science
University of Ottawa
150 Louis Pasteur, Ottawa
Ontario, K1N 6N5 Canada
{mkubat,stan}@csi.uottawa.ca

Abstract

Adding examples of the majority class to the training set can have a detrimental effect on the learner's behavior: noisy or otherwise unreliable examples from the majority class can overwhelm the minority class. The paper discusses criteria to evaluate the utility of classifiers induced from such imbalanced training sets, gives explanation of the poor behavior of some learners under these circumstances, and suggests as a solution a simple technique called one-sided selection of examples.

1 Introduction

The general topic of this paper is learning from examples described by pairs $[(\mathbf{x}, c(\mathbf{x})]$, where $\mathbf{x}$ is a vector of attribute values and $c(\mathbf{x})$ is the corresponding concept label. For simplicity, we consider only problems where $c(\mathbf{x})$ is either positive or negative, and all attributes are continuous. Since Fisher (1936), this task has received plenty of attention from statisticians as well as from researchers in artificial neural networks, AI, and ML. A typical scenario assumes the existence of a training set from which the agent induces a classifier whose performance is then assessed on an independent testing set.

An interesting complication arises when the training set is *imbalanced* in the sense that one of the classes (say the positive examples) is heavily underrepresented compared to the other class. This is encountered in many real-world applications such as the detection of fraudulent telephone calls (Fawcett and Provost, 1996); spotting unreliable telecommunications customers (Ezawa, Singh, and Norton, 1996), or a rare medical diagnosis such as the thyroid disease in the UCI repository. Extremely imbalanced classes prevail in information retrieval and filtering tasks (Lewis and Catlett, 1994).

As pointed out by many authors, the classifier's performance in applications of this kind cannot be expressed in terms of the average *accuracy* (percentage of testing examples correctly recognized by the system). In the domain studied by Lewis and Catlett (1994), only 0.2% examples are positive, and a retrieval system will achieve 99.8% accuracy by stubbornly denying the presence of the requested document. Even a system with accuracy close to 100% can thus be useless and the benefit of classifiers in similar domains must therefore be assessed by more appropriate criteria.

Informally, what the user expects is that the induced classifier will perform well on positive as well as on negative examples, rather than only on one class at the cost of the other. Section 2.1 gives a brief overview of possible options and gives reasons for the criterion that we have used in our experiments: the geometric mean, $g = \sqrt{a^+ \cdot a^-}$, of accuracies observed separately on positive examples, a^+, and on negative examples, a^-. Section 2.2 discusses the reasons why the results of learning from sparse positive examples can be disppointing (as viewed from the perspective of the g-criterion). In this way, we provide grounds for the solution described in the sequel.

In a project on detection of oil spills in satellite-borne radar images (Kubat, Holte, and Matwin, 1997) we have faced a relatively novel problem. Not only the training set is unbalanced, but the positive examples are *extremely rare:* no more than two dozens of oil slicks as compared to hundreds of lookalikes. Having observed that the g-performance of common learning systems significantly worsened as the number of negatives exceeded reasonable limits, we advocated studies of techniques tailored to this type of domains, and reported experience with the program SHRINK developed to this end (Kubat, Holte, and Matwin, 1997). The program induces classifiers in the form of a simple network of tests. The tests have the form of properly selected subintervals along the attributes' domains.

Here, we pursue quite a different strategy. If the

Table 1: Confusion matrix: the columns represent the classes assigned by the classifier; the rows represent the true classes.

		guessed:	
		negative	positive
true:	negative	a	b
	posistive	c	d

performance of the learner drops with abundant negative examples, why not simply select a reasonably sized subset of negative examples? It turns out that for our needs it is only necessary to adapt an existing technique that has been known in statistics for quite some time but seems to be by and large ignored in applications of machine learning dealing with imbalanced classes (Catlett, 1994; Pazzani et al., 1994; Fawcett and Provost, 1996; Ezawa et al, 1996; Lewis and Catlett, 1994). In particular, we adapted the technique of Tomek links (Tomek, 1976) so that it only removes examples from the majority class while leaving the examples from the minority class untouched. This is what we call *one-sided selection.* The details are presented in Section 3.

Section 4 reports experiments investigating the merits of one-sided selection in the framework of a nearest-neighbor classifier and a decision-tree generator. Experience from the oil-spill task is supplemented by experiments from other domains with similar characteristics, including three benchmark domains.

2 The Curse of Imbalanced Training Sets

2.1 Evaluation Criteria

To formulate criteria of the performance of pattern-recognition systems, statisticians work with the *confusion matrix* (Table 1) whose fields characterize classification behavior of a given system. For instance, a is the number of correctly classified negative examples and c is the number of misclassified positive examples. Performance criteria are couched in terms of these numbers: the traditional accuracy (inappropriate for our needs) is calculated as $acc = \frac{a+d}{a+b+c+d}$.

The information-retrieval community prefers to work with *precision*, $p = \frac{d}{b+d}$, and *recall*, $r = \frac{d}{c+d}$. Sometimes, the geometric mean of the two quantities is used, $\sqrt{p \cdot r}$, reaching high values only if both precision and recall are high and in equilibrium. Other authors (Lewis and Gale, 1994) combine precision and recall into a more elaborate function called the F-measure.

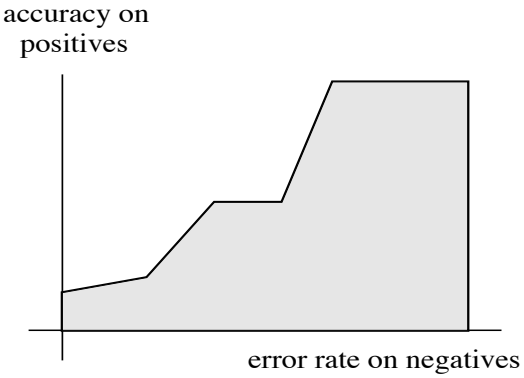

Figure 1: The ROC curve

Yet another useful measure is the information-based criterion suggested and analyzed by Kononenko and Bratko (1991).

Swets (1988) discusses a measure that reflects the fact that very often the classifier can deliberately be biased towards one of the classes and that the extent of this bias can be controled: accuracy on positive examples, $a^+ = \frac{d}{c+d}$, can be increased at the cost of accuracy on negative examples, $a^- = \frac{a}{a+b}$. The relation of the two quantities can be captured by what is called the ROC (Relative Operating Characteristics) curve: the horizontal axis measures error rate on negative examples while the vertical axis measures accuracy on the positive examples. An example is shown in Figure 1. Informally, the curve shows to what extent accuracy on positive examples drops with reduced error rate on negative examples. The larger the area below the ROC curve, the higher the classification potential of the system.

In Kubat, Holte, and Matwin (1997) we used the geometric mean of the accuracies measured separately on each class: $g = \sqrt{a^+ \cdot a^-} = \sqrt{\frac{a}{a+b} \cdot \frac{d}{c+d}}$. This measure relates to a point on the ROC curve and the idea is to maximize the accuracy on each of the two classes while keeping these accuracies *balanced.* For instance, a high a^+ by a low a^- will result in poor g.

This measure was consistent with the requirements of the customer in the oil-spill domain. However, the concrete choice does not seem to be critical for the considerations in the sequel, and we believe that other metrics (i.e. metrics sensitive to the needs of each class) could do as well. Especially the criterion discussed at length by Kononenko and Bratko (1991) deserves attention for its many convenient properties.

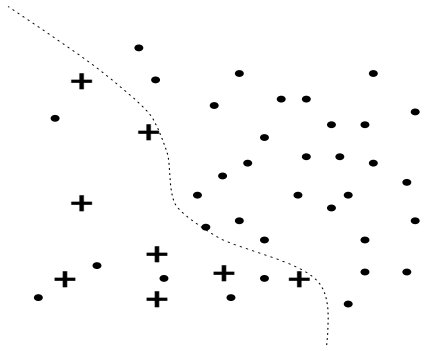

Figure 2: The sparseness of positive examples complicates the learner's task

2.2 The Case of Extremely Rare Positives

Why do abundant negatives hurt? Figure 2 offers an intuition that explains the behavior of the nearest-neighbor rule (1-NN) under these circumstances. The picture shows several positive examples, many negative examples, and a decision surface that is a priori unknown to the learner. The reader can see that each positive example has a negative nearest neighbor. This can be generalized: as the number of negative examples in a noisy domain grows (the number of positives being constant), so does the likelihood that the nearest neighbor of *any* example will be negative. Many positive examples will thus be misclassified. With infinitely many negative examples and a finite number of sparse positive examples, the 1-NN classifier experiences $a^- = 100\%$ and $a^+ = 0\%$, which is unacceptable. With reasonably sized training sets, this situation can be partially rectified by taking k nearest neighbors instead of a single one. Still, with very large training sets (and with very large disproportion between the positive and negative examples), the harmful effect of the negative examples prevails.

Similar conclusions can be inferred from the principles of Bayesian classifiers. Denote by $P(+)$ and $P(-)$ the a priori probabilities of the positive and negative class, respectively, and denote by $p_+(\mathbf{x})$ and $p_-(\mathbf{x})$ the probability density functions for the positive and negative class at the point $\mathbf{x}$. The pure Bayesian classifier (ignoring miclassification costs) labels $\mathbf{x}$ as a positive example if $P(+)p_+(\mathbf{x}) > P(-)p_-(\mathbf{x})$. However, the inequality can hardly ever be satisfied in the learning task depicted in Figure 2 because $P(-) \gg P(+)$ and only rarely, if ever, $p_+(\mathbf{x}) > p_-(\mathbf{x})$. The only solution is to allot very high *costs* to positive examples (l_+) by a small cost associated with negative examples (L_-). The classifier would then assign the positive class whenever $P(+)p_+(\mathbf{x})l_+ > P(-)p_-(\mathbf{x})l_-$ Even then, the classifier can have problems to properly estimate a smooth density function of the positive class. Moreover, it is often not clear how to determine the values of l_+ and l_- during learning or prior to learning.

Induction of decision trees will suffer as well. Decision trees are known to be universal classifiers: any dichotomy of points in general position in an n-dimensional continuous space can be realized by a sufficiently large decision tree. In the case from Figure 2, each positive example will eventually be represented by one branch of the tree. With sufficiently many negative examples and sparse positive examples, the positive regions will be arbitrarily small. The tree overfits the data with a similar effect as in the case of the 1-NN classifier.

Pruning the tree does not answer the main problem. After pruning, some regions will contain examples from both classes. A commonplace policy is to associate with each leaf the label of the class that has majority in the corresponding region. In applications with rare positive examples, regions containing mixed positives and negatives will be labeled as negative, with the potential effect that none of the branches will be deemed positive unless the algorithm is appropriately modified. Of course, the situation can be improved by meticulously finding the best pruning constant, say, by means of a properly designed cross-validation technique, as in CART (Breiman et al., 1984. However, this does not address the core of the problem: each positive example is surrounded by one or more (or even many) negative examples, and most regions will thus be labeled as negative.

3 One-Sided Sampling

Learning from highly imbalanced training sets received some attention in the neural-network community. Common solutions duplicate the training examples; create new examples by corrupting existing ones with artificial noise; or increase the learning rate when an example of the underrepresented concept is presented (DeRouin et al. 1991). In the realm of machine learning, the problem has been addressed in various ways: by weighing training instances (Pazzani et al. 1994), by introducing different miclassification costs for positive and negative examples (Gordon and Perlis (1989), by windowing and bootstrapping (Catlett, 1991; Sung and Poggio, 1995), by heterogeneous sampling (Lewis and Catlett, 1994), and by forcing the learner to focus on specific relationships between certain attributes (Ezawa et al, 1996). Note that this problem is different from the pure scarcity of data as discussed in Dietterich, Lathrop, and Lozano-Perez (1997).

As already mentioned, the strategy chosen in this particular study is that the learner will first select a repre-

sentative subset of the negative examples. The training set becomes more balanced, and the drawbacks discussed in the previous section will diminish.

Selection techniques were studied by the statistical literature of the 60s and 70s (Hart, 1968; Gates, 1972; Tomek, 1976) and were later addressed also by machine-learning researchers, among them by Aha, Kibler, and Albert (1991), Zhang (1992), Skalak (1994), Lewis and Gale (1994), Floyd and Warmuth (1995). Although the focus of these papers was mainly on the reduction of the training-set *size,* the underlying algorithms can be instrumental also in our particular problem. The only requirement is that the learner always keeps all positive examples (they are too rare to be wasted, even under the danger that some of them are noisy) and prunes out only negative examples.

What heuristics can be applied to detect less reliable examples? Figure 2 has already illustrated the fact that negative examples can roughly be divided into four groups.

1. Those that suffer from the *class-label noise*—for instance the point in the bottom left corner.
2. *Borderline* examples that are close to the boundary between the positive and negative regions. Borderline examples are unreliable: even a small amount of attribute noise can send the example to the wrong side of the decision surface.
3. Those that are *redundant* so that their part can be taken over by other examples. This is the case of examples in the upper right corner.
4. *Safe* examples that are worth being kept for future classification tasks.

The redundant examples do not harm correct classifications but they increase classification costs. Figure 3 shows what happens with the training set from Figure 2 if all borderline and noisy examples are removed. Figure 4 illustrates a further reduction: the removal of redundant negative examples. If examples from this last picture are used as a training set, neither the 1-NN rule nor a bayesian classifier nor a decision-tree generator should run into any serious problem.

An intelligent agent will thus try to eliminate borderline examples and examples suffering from the class-label noise. These can easily be detected using the concept of *Tomek links* (Tomek, 1976). The idea can be put as follows. Take two examples, $\mathbf{x}$ and $\mathbf{y}$, so that each has a different concept label. Denote by $\delta(\mathbf{x},\mathbf{y})$ the distance between $\mathbf{x}$ and $\mathbf{y}$. The pair $(\mathbf{x},\mathbf{y})$ is called a Tomek link if no example $\mathbf{z}$ exists such that $\delta(\mathbf{x},\mathbf{z}) < \delta(\mathbf{x},\mathbf{y})$ or $\delta(\mathbf{y},\mathbf{z}) < \delta(\mathbf{y},\mathbf{x})$. Examples participating in Tomek links are either borderline or noisy.

An attempt to reduce the number of redundant examples can be cast as the task of creating a *consistent subset*, C, of the training set, S. By definition, a set

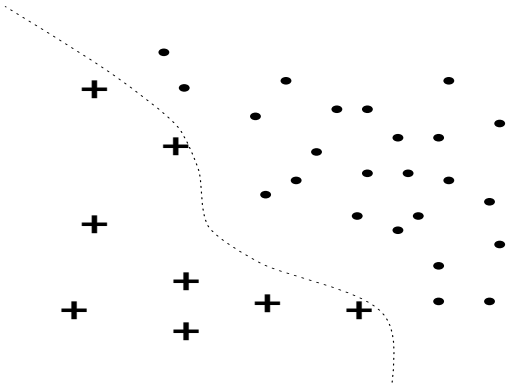

Figure 3: The training set without the borderline and noisy negative examples.

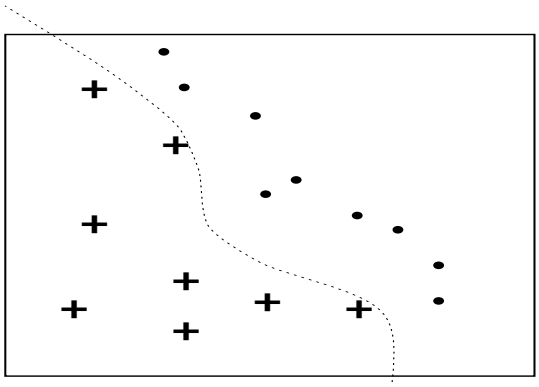

Figure 4: The training set after the removal of redundant negative examples.

Table 2: Algorithm for the one-sided selection of examples.

1. Let S be the original training set.
2. Initially, C contains all positive examples from S and one randomly selected negative example.
3. Classify S with the 1-NN rule using the examples in C, and compare the assigned concept labels with the original ones. Move all misclassified examples into C that is now consistent with S while being smaller.
4. Remove from C all negative examples participating in Tomek links. This removes those negative examples that are believed borderline and/or noisy. All positive examples are retained. The resulting set is referred to as T.

$C \subseteq S$ is consistent with S if, when used by the 1-NN rule, it correctly classifies examples in S. Note that any training set is a consistent subset of itself. In our particular problem, the objective is not necessarily to create the smallest possible C. Rather, it is enough if the set of negative examples reasonably shrinks. To this end, we use our own variant of the technique invented by Hart (1968). To start with, one negative and all positive examples are placed into C. Then, the 1-NN rule is used with the examples in C in an attempt to re-classify S. Those training examples that have been misclassified are then added to C.

Table 2 summarizes the procedure. First, the number of redundant negatives is reduced by creating the subset C, consistent with the training set. Then, the system removes those negative examples that participate at Tomek links. In this way, the noisy and borderline examples are discarded, which leads to the new training set, T.

4 Experiments

4.1 Experimental Setting

The task of the experiments is to demonstrate that in applications with imbalanced training sets, one-sided sampling indeed improves the behavior of some existing learners. In particular, 1-NN and C4.5 were selected because of their widespread use and well-known performance. We wanted to compare the results of a classifier induced from *all* training examples (the set S) with the results of the same classifier induced from the sets C (some redundant examples removed) and T (some redundant, borderline, and noisy examples removed).

To obtain statistically reliable results even under the circumstance of scarce positive examples, a specific variant of the k-fold cross-validation technique was used. The training set was divided into k subsets of equal size in a manner that ensured that each of them had the same proportion of positive and negative examples (*stratified sampling*). For all possible choices of $k-1$ subsets, the union of $k-1$ subsets was used for training, and the induced classifier was tested on the remaining subset. The results were averaged.

The domains used as experimental testbeds can roughly be divided into two groups. The first group is formed by data from two major projects that originally motivated this research: oil-spill detection and sleep classification. They represent very difficult learning tasks because the information provided by the given attributes is insufficient for successful classification. Moreover, many attributes are probably irrelevant.

Two data files come from the oil-spill detection project reported by Kubat, Holte, and Matwin (1997). Each of them actually refers to the same set of satellite images, each time preprocessed by a different image processing method. The positives are rare, and the domains perfectly fit the problem definition from Section 1. Another two data files, kr and br, come from an earlier research of one of the authors (Kubat, Pfurtscheller, and Flotzinger, 1994). The training set is imbalanced but the number of positive examples is not so small. The negative examples are known to belong to several different subclasses.

Table 3: Characterization of experimental data. All attributes are continuous and each domain has only two classes.

file	# attrib.	#pos	#neg	k
oil1	44	24	480	8
oil2	39	21	350	7
kr	15	150	750	5
br	15	140	700	5
g7	10	28	182	7
vw0	10	90	900	5
veh1	19	168	676	4

Apart from these domains, we also followed the common practice of the machine-learning community and experimented with some data files from the UCI repository (Murphy and Aha, 1994) so that other researchers can replicate the experiments and double-check our results. To adapt the data to our needs, we defined the task as learning to distinguish one selected class from the other classes. More specifically, in the glass domain (g7), the task is to learn class 7; in the vowels domain (vw0), the task is to learn class 0; and in the vehicles domain (veh1), the task is to learn class 1.

The testbeds are summarized in Table 3. For each domain, the table gives the number of attributes, the number of positive examples, the number of negative examples, and the value of k defining the k-fold cross-validation technique. In two benchmark domains (g7, veh1), we discarded (randomly) some examples just to ensure that each of the k subsets will contain exactly the same proportion of positive and negative examples (which can only be ensured if if the number of positive examples and the number of negative examples can both be divided by the same integer number k).

4.2 Results and Discussion

In downsizing the training set, we delete all examples that participate in Tomek links. We do not re-

move all redundant examples because this could be prohibitively expensive. The algorithm described in Section 3 is relatively cheap, and capable of removing most of the redundant examples. Preliminary experiments revealed that the performance of the induced classifier is largely unaffected by the choice of redundant examples to be removed.

The results are summarized in Tables 4 through 7. Each table is divided into three parts. The first, S, gives results achieved when using *all* training examples; the second part, C, pertains to the situation after the removal of redundant negative examples; and the third part, T, pertains to the case when the system removed from C all negative examples participating in Tomek links. The values in the tables are obtained from the k-fold crossvalidation technique (for the values of k in the individual domains see Table 3).

To better illustrate the learners' behavior, the tables give the results in terms of the accuracy on positive examples ($a^+ = \frac{d}{c+d}$), accuracy on the negative examples ($a^- = \frac{a}{a+b}$), and the geometric means of these two values: $g = \sqrt{a^+ \cdot a^-}$. The reader can see that the removal of redundant negative examples, while significantly reducing the number of stored examples, does not yet guarantee performance gain. Indeed, in some cases (1-NN in oil2 and C4.5 in oil1) the g-performance even dropped because the removal of redundant negatives did not solve the main problem: poor accuracy on positive examples. The abundant negative examples in the borderline region bias the classifier towards the negative class. Once these examples are removed (set T), the accuracy as measured on different classes of examples (positive and negative) becomes more balanced, and the value of g improves in each of the four domains, sometimes even by a wide margin. The extreme is the behavior of C4.5 in the oil-slick-II domain where the difference between the g-performance of the S set and that of the T set is more than 16%, and the behavior of 1-NN on the oil-slick-I domain where the improvement is even 46%.

For comparison, the last columns in the tables give the average accuracy, $acc = \frac{a+d}{a+b+c+d}$. The reader can see that the values of acc do not express anything alarming. The accuracy achieved by the S set is sometimes even higher than that of T. And yet the values of $a+$ indicate that the utility in a real-world setting might be dubious: for instance, missing 80% of the positive examples in the oil-slick-I domain.

Table 8 summarizes the g-performance for the benchmark domains, indicating also the limitations of the technique. In the vehicles domain, both 1-NN and C4.5 significantly profited from the one-sided sampling. However, in the glass domain the sampling technique leads only to a modest improvement in 1-NN

Table 4: The results in oil-slicks I

	#ex.	progr.	g	a^+	a^-	acc
S	441.0	1-NN	44.3	20.8	94.4	90.9
		C4.5	82.9	72.0	95.5	94.4
C	83.0	1-NN	66.6	45.8	96.7	94.3
		C4.5	79.1	66.7	93.8	92.5
T	65.2	1-NN	90.6	87.5	93.7	93.4
		C4.5	84.3	79.2	89.8	89.3

Table 5: The results in oil-slicks II

	#ex.	progr.	g	a^+	a^-	acc
S	318.0	1-NN	51.3	28.6	92.3	88.7
		C4.5	49.5	28.6	85.7	82.5
C	119.9	1-NN	41.4	19.0	90.0	86.0
		C4.5	56.5	42.9	74.6	72.8
T	115.3	1-NN	53.0	33.3	84.3	81.4
		C4.5	66.0	57.1	76.3	75.2

Table 6: The results in KR

	#ex.	progr.	g	a^+	a^-	acc
S	720.0	1-NN	69.2	52.7	90.9	84.5
		C4.5	74.0	59.3	92.3	86.8
C	375.2	1-NN	69.8	55.3	88.0	82.6
		C4.5	75.3	62.0	91.5	86.6
T	267.4	1-NN	75.8	74.0	77.6	77.0
		C4.5	80.8	78.0	83.6	82.7

Table 7: The results in BR

	#ex.	progr.	g	a^+	a^-	acc
S	672.0	1-NN	81.2	70.7	93.3	89.5
		C4.5	76.4	62.1	94.0	88.7
C	297.2	1-NN	81.3	72.9	90.7	87.7
		C4.5	79.0	69.3	90.1	86.6
T	227.2	1-NN	87.8	93.6	82.4	84.3
		C4.5	83.4	80.7	86.1	85.2

Table 8: Results observed in the benchmark domains

	g7		vw0		veh1	
	1-NN	C4.5	1-NN	C4.5	1-NN	C4.5
S	95.2	92.6	83.4	88.4	52.1	57.6
C	96.6	92.6	90.8	84.1	55.4	62.0
T	96.6	84.5	90.9	84.0	66.8	69.4

while causing a performance drop in C4.5. A more detailed examination revealed that in this domain, C4.5 did *not* yield disproportionate values of a^+ and a^-, and the situation therefore did not call for one-sided sampling. Similar behavior was observed also in the vowels domain where only 1-NN (and not C4.5) experienced improvement.

This suggest how the algorithm should actually be applied. First, look whether the values of a^+ and a^- are balanced. Only if one of them is prohibitively low, carry out the one-sided sampling.

5 Conclusion

In some real-world tasks, the learner can avail itself of just a few positive examples and virtually unlimited number of negative examples. If this is the case, the avarage classification accuracy on the testing set is not a very useful criterion. Several alternative criteria can be recommeded, from which we selected the g-performance: the geometric mean of a^+ (percentage of positive examples correctly recognized) and a^- (percentage of negative examples correctly recognized).

Common learning algorithms such as the nearest-neighbor rule and induction of decision trees can be misled when the number of negative examples exceeds certain limits. This behavior pertains to the fundamental principles of these learners, as explained in the paper. The sensitivity to an imbalanced distribution of examples can be mitigated by selection techniques that discard those negative examples that lie in the borderline region, are noisy, or redundant.

The paper investigated the impact of simple selection techniques, adapted so that they remove only negative examples while keeping all positives. Hence the name: *one-sided selection*. Although the experiments addressed only 2-class problems, we believe that a similar approach can be used also in the frame of multi-class learning.

Acknowledgements

The research was partially supported by Precarn, Inc. Thanks are due to Rob Holte for his many useful comments on early versions of this paper. The sleep data used in this research belong to the Department of Medical Informatics, Technical University in Graz, Austria, and have been recorded and classified under a grant from the 'Fonds zur Förderung der wissenschaftlichen Forschung' (Project S49/03). Thanks are due to Gert Pfurtscheller for his kind permission to use these data.

References

Aha D., Kibler D., and Albert M.K. (1991). Instance-Based Learning Algorithms. *Machine Learning*, 6(1), 37–66

Breiman, L., Friedman, J., Olshen, R., and Stone, C.J. (1984). *Classification and Regression Trees.* Wadsworth International Group, Belmont, CA

Catlett, J. (1991). Megainduction: A Test Flight. *Proceedings of the 8th International Workshop on Machine Learning* (pp.596–599), San Mateo, CA: Morgan Kaufmann

DeRouin, E., Brown, J., Beck, H., Fausett, L, and Schneider, M. (1991). Neural Network Training on Unequally Represented Classes. In Dagli, C.H., Kumara, S.R.T. and Shin, Y.C. (eds.): *Intelligent Engineering Systems Through Artificial Neural Networks*, ASME Press, New York, 135–145

Dietterich, T.G., Lathrop, R.H., and Lozano-Perez, T. (1997). Solving the Multiple-Instance Problem with Axis-Parallel Rectangles. to appear in *Artificial Intelligence*

Ezawa, K.J., Singh, M. and Norton, S.W. (1996). Learning Goal Oriented Bayesian Networks for Telecommunications Management. *Proceedings of the International Conference on Machine Learning, ICML'96* (pp. 139–147), Bari, Italy, Morgan Kaufmann

Fawcett, T. and Provost, F. (1996). Combining Data Mining and Machine Learning for Effective User Profile. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 8–13), Portland OR, AAAI Press

Floyd, S. and Warmuth, M. (1995). Sample Compression, Learnability, and the Vapnik-Chervonenkis Dimension. *Machine Learning*, 21, 269–304

Gates, G.W. (1972). The Reduced Nearest Neighbor Rule. *IEEE Transactions on Information Theory*, 18, 431–433

Gordon, D.F. and Perlis, D. (1989). Explicitly Biased Generalization. *Computational Intelligence*, 5, 67–81

Hart, P.E. (1968). The Condensed Nearest Neighbor Rule. *IEEE Transactions on Information Theory*, IT-14, 515–516

Kononenko, I. and Bratko, I. (1991). Information-Based Evaluation Criterion for Classifier's Performance. *Machine Learning*, 6, 67–80

Kubat, M., Holte, R., and Matwin, S. (1997). Learning when Negative Examples Abound. *Proceedings of the 9th European Conference on Machine Learning, ECML'97*, Prague

Kubat, M., Pfurtscheller, G., and Flotzinger D. (1994).

AI-Based Approach to Automatic Sleep Classification. *Biological Cybernetics*, 79, 443–448

Lewis, D. and Catlett, J. (1994). Heterogeneous Uncertainty Sampling for Supervized Learning. *Proceedings of the 11th International Conference on Machine Learning, ICML'94* (pp. 148–156), New Brunswick, New Jersey, Morgan Kaufmann

Lewis, D. and Gale, W. (1994). Training Text Classifiers by Uncertainty Sampling. *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*

Murphy, P. and Aha, D. (1994). UCI Repository of Machine Learning Databases [machine-readable data repository]. Technical Report, University of California, Irvine

Pazzani, M, Merz, C., Murphy, P., Ali, K., Hume, T., and Brunk, C. (1994). Reducing Misclassification Costs. *Proceedings of the 11th International Conference on Machine Learning, ICML'94* (pp. 217–225), New Brunswick, New Jersey, Morgan Kaufmann

Quinlan J.R. (1993). *C4.5: Programs for Machine Learning.* Morgan Kaufmann, San Mateo

Skalak, D. (1994). Prototype and Feature Selection by Sampling and Random Mutation Hill Climbing Algorithms. *Proceedings of the 11th Machine Learning Conference* (293–301), New Brunswick, Morgan Kaufmann

Sung, K-K. and Poggio, T. (1995). Learning Human Face Detection in Cluttered Scenes. *Proceedings of the 6th International Conference on Computer Analysis of Images and Patterns*, Prague

Swets, J.A. (1988). Measuring the Accuracy of Dignostic Systems. *Science*, 240, 1285–1293

Tomek I. (1976). Two Modifications of CNN. *IEEE Transactions on Systems, Man and Communications*, SMC-6, 769–772

Zhang, J. (1992). Selecting Typical Instances in Instance-Based Learning. *Proceedings of the 9th International Machine Learning Workshop* (pp. 470–479), San Mateo, CA, Morgan Kaufmann