Research Issues Arising in Applying Machine Learning to Oil Slick Detection

Miroslav Kubat
Center for Advanced Computer Studies
University of Southwestern Louisiana
P.O. Box 44330, Lafayette, LA 70504-4330
and Department of Computer Science
Southern University
Baton Rouge, LA 70813-0400
mkubat@cacs.usl.edu

Robert Holte, Stan Matwin
School of Information Technology and Engineering
University of Ottawa
150 Louis Pasteur, Ottawa
Ontario, K1N 6N5 Canada
{holte,stan}@site.uottawa.ca

Abstract

Applications in image processing and remote sensing raise questions that have so far received only marginal attention from the machine learning community. And yet, each of our issues, we believe, represents a rich research topic, in addition to being of practical importance. The task of the presented case study is to expose some of them in the context where we encountered them, and to encourage discussion that might spawn further research.

Introduction

Our intention is to briefly report on some of our experience from a major project that employed machine learning techniques for the recognition of oil spills in satellite-borne radar images of the sea surface. The work is now finished, and its product is being field tested and marketed. The main observations made during the research have been published in two conference papers (Kubat, Holte, and Matwin, 1997; Kubat and Matwin, 1997) and the technical results are detailed in a full-length journal paper (Kubat, Holte, and Matwin, 1998). Nevertheless, as is often the case, some of the issues turned out to be weak and/or insufficiently documented to enter an article in a leading journal. Still, they deserve to be discussed, and this workshop is surely the right forum. To raise these issues at this workshop is important to us because we felt that some vital problems that regularly accompany major applications have so far been insufficiently addressed in the literature.

As already indicated, a detailed description and motivation of the learning task, together with the attendant technical results, are the subject of Kubat, Holte, and Matwin (1998). The objective was to develop a tool that would learn to detect oil spills in radar images. Oil spills appear as dark regions in a radar image. Unfortunately, so do several commonly occurring natural phenomena, such as wind slicks (winds with speeds exceeding 10m/sec decrease the reflectance of the radar beam, hence the affected area looks darker in a radar image), rain, algae, plankton, etc. These are called "lookalikes," and the main challenge in detecting oil spills is to distinguish the oil spills from the lookalikes.

Image processing techniques are used to normalize the image in certain ways (e.g. to correct for the radar beam's incidence angle), to identify suspicious dark regions, and to extract features (attributes) of each region that can help distinguish oil spills from lookalikes. The list of the dark regions then forms the training set. Oil spills are treated as positive examples, and the lookalikes as negative. The examples are described by dozens of attributes such as size, average brightness, length-vs-breadth ratio, average sharpness of the edges, "jaggedness" of the edges, etc. The image processing part was developed by Macdonald Dettwiler Associates, a company specializing in remote sensing. The input of image processing is thus a raw image, the output being a set of fixed length attribute vectors, one for each suspicious region (if the system failed to discover any dark region in the image, no new vectors are output). During normal operation, the vectors are fed into a classifier to decide which image, and which regions within an image, are to be presented for human inspection for the final decision.

The classifier is created by a learning algorithm whose development was the main task of our project. Prior to learning, the examples (regions) are classified by a human expert as oil slicks and lookalikes, but these classifications are, admittedly, imperfect: on some occasions, the expert was unsure whether or not a particular region was an oil slick. The class labels can thus be erroneous. The learner outputs a classifier capable of deciding, with a certain degree of reliability, whether a specific dark region is an oil spill.

In the sequel, we focus on three classes of problems. First, those issues that are encountered at the time of problem specification are discussed in Section 2. They include the decision whether to deliver a classifier or a user-tailored learner, and the question of the appropriate granularity of the data. Section 3 then discusses some general data characteristics that are likely to be encountered also in other similar domains: imbalanced distribution of positive and negative examples, and the fact that the training set is substructured into small batches. Section 4 summarizes the experiences.

Problem-Specification Issues

Prior to beginning the development, the designer of a machine learning system has to answer a few vital questions related to the way the product is to be employed, and to the nature of the learner's, and classifier's, input. Let us discuss these issues in following two subsections.

A Classifier or a Learner?

¿From the user's point of view, machine learning can in principle be employed in two different ways. More commonly, the customer is provided with a classifier. A host of fielded applications falling into this category is summarized by Langley and Simon (1998). A machine learning specialist receives a set of classified training examples, and is asked to deliver a tool, say, a decision tree, that will be used to classify future examples.

In our project, an alternative task was chosen: to develop a learning program, rather than a mere classifier. Why would a user prefer this option? The answer is simple. In our domain, each end user will deal with a somewhat different type of oil slicks whose characteristics will depend largely on various geographical factors (climate, vicinity of land mass, abundance of plankton, etc.) and specific circumstances such as the proximity of oil platforms. These factors and circumstances are virtually unknown at the time of the classifier development. Moreover, since the data available for learning is limited, the relevant examples for learning are simply not available. It is well known that when a classifier has been trained in one context, and then applied in another context, the classification performance can significantly drop, as discussed at one of th recent ICML'96 workshops (Kubat and Widmer, 1996). Given this situation, a skeptic can rightly ask why to employ machine learning at all. Why not simply take some of off-the-shelf learning product and train it on the data?

It is true that we did not have enough data. However, from what we had we could at least make sound judgements about the kind of data that will typically characterize oil-spill learning. For instance, the training set will nearly always be imbalanced: among the many training examples, only a few will be positive—this is due to the high price of the images, and by the relative scarcity of oil spills in these images. Second, the examples will be described by dozens of attributes, from which only few will be relevant (although different sets of attributes might be relevant in different contexts). Finally, the attributes differ in ranges and scales (e.g. some can of them give numbers of pixels, others are in decibels). Given that these are very likely the characteristics of the data encountered by the system's users, we decided to develop a learning system tailored to these features. Although we did not have as many examples as we would like, we could side-step this deficiency by resorting to those benchmark data that are known to have similar characteristics.

There is one more reason for delivering a learner rather than a classifier. At the beginning, the user does not have enough data, but still wants a classifier and is willing to tolerate its imperfectness. In the future, more examples will become available, and these examples can be incorporated in a new, much larger, training set. The learning process can then be repeated, and a better classifier obtained.

The idea of deploying a learner, rather than a classifier, is far from being common in the literature. As one of the rare exceptions, Armstrong et al. (1998) deserves to be cited. In their case, a WWW-user provides examples of articles to be filtered out from the many available electronic journals.

Choosing the Right Granularity

An important decision in learning to identify objects in images concerns *granularity.* Essentially, three different approaches come into question. The first works with the whole image, and its output simply states whether the given image contains an oil slick. The second approach works with the dark regions detected in the images, and provides the user with their coordinates. The third approach classifies individual pixels ("this pixel is part of an oil slick"), as has been done, for instance by Ossen et al (1994). The approach operating with pixels represents the finest granularity, whereas the approach operating with the images represents the coarsest granularity.

The choice of granularity heavily affects the memory requirements, computational demands, as well as reliability. For one thing, finer granularity yields more examples. In our project we used 9 images that contained about one thousand dark regions. However, these images contained several millions pixels. Obviously, millions of examples are not as easy to handle as hundreds of regions or just a few images. But then, with finer granularity, a higher misclassification rate can be tolerated: an image containing several regions considered as positive with only moderate confidence will be classified as "slick-containing" even if none of the regions is suspicious enough to be individually classified as an oil spill. On the other hand, single pixels can only be described with a substantially impoverished set of features. After all, it is the relations among pixels that contain most of the information. Another unpleasant aspect of working with single pixels is that the result will not look coherent—there is no guarantee that the "oil slick" pixels will form coherent regions in an image.

For all these reasons we decided that our system would work with regions as detected by the image-processing subsystem. It is important to bear in mind that this means to increase the burden imposed on the image processing specialists. In our case this worked fine because our partner was a leading company in the remote-sensing industry. However, in a "normal" university environment, the effort related to the description of regions can easily become a prohibitive factor.

The issue of the degree of granularity arises in many applications. For instance, in semiconductor manufacturing (Turney, 1995), circuits are manufactured in

Table 1: The numbers of positive and negative examples in the images

im.	1	2	3	4	5	6	7	8	9	all
pos.	8	4	2	6	2	4	3	5	7	41
neg.	3	180	101	129	60	70	76	80	197	896
total	11	184	103	135	62	74	79	85	204	937

batches of wafers, and the system can be required to classify an entire batch, wafer, or to operate at even lower levels. Likewise, the text-to-speech mapping discussed by Dietterich et al. (1995) can be addressed in four distinct levels of granularity.

Data Characteristics

Table 1 summarizes a typical data set for learning. As a matter of fact, we experimented with many such sets, but for reasons irrelevant to this study, these sets could not be combined. The reader should understand the table as a good illustration of typical training data. Two observations can be made. First, there are relatively few positive examples, heavily outnumbered by the negative examples. Second, the data come in small batches.

Scarcity of Positive Examples

The first aspect is the *scarcity* of positive examples. Although satellites are continually producing images, most of these images contain no oil spills, and we did not have access to an automatic system for identifying those that do. A human expert had to view each image, detect suspicious regions (if there we any), and classify these regions as positive and negative. In addition to the genuine infrequency of oil spills and the limited time the expert was willing (and able) to spend on the data preparation, the amount of data was restricted by financial considerations: images cost hundreds, sometimes thousands of dollars each. For our experiments it was decided to purchase a set of 9 carefully selected images that were known to contain oil spills. The situation with scarce positive examples is quite common in real-world applications. For example, in the drug activity application reported by Dietterich et al. (1997) the two datasets contain 47 and 39 positive examples respectively.

Moreover, we deliberately worked with training sets whose distribution of positive and negative examples did not truly reflect the actual domain. In reality, only a fairly small percentage of images contain oil spills. However, as the training images had to be purchased for a relatively high price, economic constraints forced us to concentrate only on images that provably contained positive examples (otherwise, given the maximum examples that we could purchase, the number of positive examples would be even smaller). This means that in reality the ratio between oil spills and lookalikes is supposed to be lower that during the training, which only adds to the importance of the considerations in the next section.

As a final note, the number of positive and negative examples was far from being fixed. The image-processing system that was used to detect the dark regions contained several parameters whose actual setting could substantially affect the size of the training set. For instance, one parameter determined whether a region was dark enough to be considered as an example. This threshold was able to affect the number of examples by an order of magnitude. Another parameter determined the minimum size (in pixels) of a region.

The lesson is that in major applications of machine learning the designer will often have a control of the contents of the training set, which is somewhat at variance with the common practice to work with fixed and immutable training sets. Frankly, the lack of this experience at the beginning of our project caused that we largely underestimated the consequences of the need to use machine intelligence to finalize the training set. Tailoring of the training set perhaps should have been made an integral part of the learning process.

Imbalanced Training Sets

Another characteristic of the oil spill domain is that there were many more negative examples (lookalikes) than positive examples (oil slicks)—the majority class usually represented more than 95% of the data. Again, this is, we think, by no means an exceptional situation. Highly imbalanced training sets occur in applications where the classifier is to detect a rare, though important event, such as fraudulent telephone calls (Fawcett and Provost, 1996), unreliable telecommunications customers (Ezawa, Singh, and Norton, 1996), failures or delays in a manufacturing process (Riddle, Segal, and Etzioni, 1994), or rare diagnoses (e.g. the thyroid diseases in the UCI repository (Murphy and Aha, 1994)). Extremely imbalanced classes also arise in information retrieval and filtering tasks: in the domain studies by Lewis and Catlett (1994), only 0.2% (1 in 500) examples are positive.

An important implication of the fact that the classes are unevenly distributed is the fact that the most popular performance metric, the classification accuracy, loses much of its appeal. To see why, consider the case where the relative frequency of negative examples is 96%. A classifier that labels all regions as negatives will achieve the seemingly impressive accuracy of 96%. And yet it would be useless because it totally fails to achieve the fundamental goal: to detect oil spills. A system achieving 94% on spills and 94% on non-spills will have a worse overall accuracy in spite of being deemed as highly successful. This indicates that a proper choice of the performance metric deserves particular attention.

In many applications, the accuracy is defined using the confusion matrix from Table 2 as $acc = \frac{a+d}{a+b+c+d}$. In other words, accuracy is the percentage of examples correctly classified. Informally, we want to present to

Table 2: Confusion matrix

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

the user as many spills as possible provided that the total number of false alarms is not too large. Curves used to visualize the tradeoff between these two requirements are called ROC curves (Swets, 1988). Figure 1 shows a typical ROC curve obtained on the oil-spill data with the system described by Kubat, Holte, and Matwin (1998). It is a plot with the percentage of correctly classified positive examples ($\frac{d}{c+d}$) on the y-axis and the false positive rate ($\frac{b}{a+b}$) on the x-axis. The perfect classifier corresponds to the point (0,100): 0 false positives (i.e., 0% error on the negative examples) and 100% accuracy on the positive examples. The extreme points of the curve, (0,0) and (100,100), correspond to classifiers that classify all examples as negative and positive, respectively.

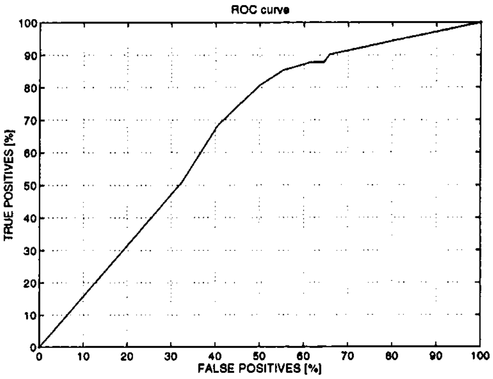


Figure 1: An ROC curve

To measure performance in environments with imbalanced classes, the information retrieval community works with *recall* ($r = \frac{d}{c+d}$) and *precision* ($p = \frac{d}{b+d}$) and combines them by way of a geometric mean ($\sqrt{r \cdot p}$) or the more sophisticated F-measure (Lewis & Gale, 1994). Other measures have been suggested (van Rijsbergen, 1979, Chapter 7), including an information theoretic formula suggested by Kononenko and Bratko (1991).

We decided that in the version of the system that will be delivered to end users there will not be a preprogrammed way of condensing the ROC curve to a single performance measure. Instead, the user will be able to move along the curve and choose the point that best meets his/her current needs. In this way, the user perceives the performance in terms of two parameters (the frequency of true positives and of false positives). This is typical of fielded systems. As pointed out by Saitta, Giordana and Neri (1995), systems that serve as tools for users confronting a specific decision (e.g., whether to send an aircraft to verify a spill and document the incident) should not be constrained to use a scalar performance measure. The user needs to be able to tune the system's behavior so as to trade off various conflicting needs.

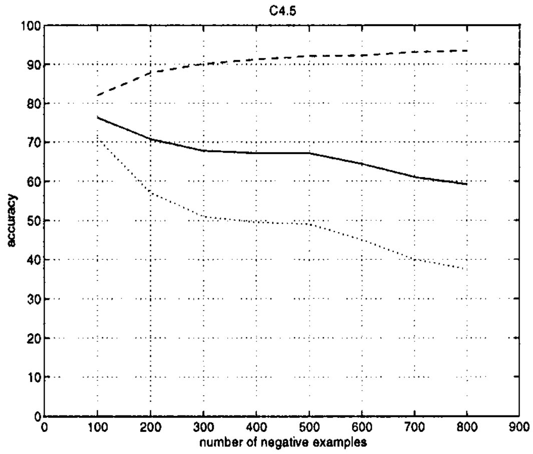
Although, in general, the challenge is to build a system that can produce classifiers across a maximally broad range of its ROC curve, in the course of development we did not have access to the users that would tune the system to their particular circumstances. However, we needed a performance measure to provide immediate feedback (in terms of a single value) on our design decisions. This measure would have to address the clear inadequacy of accuracy, which is unusable in our problem. To this end, we have mainly used the geometric mean (*g-mean*), $g = \sqrt{acc{+} \times acc{-}}$, where $acc{+} = \frac{d}{c+d}$ is the accuracy on the positive examples, and $acc{-} = \frac{a}{a+b}$, is the accuracy on the negative examples. This measure has the distinctive property of being independent of the distribution of examples between classes, and is thus robust in circumstances where this distribution might change with time or be different in the training and testing sets. Another important and distinctive property is that *g-mean* is nonlinear. A change of p percentage points in *acc+* (or *acc−*) has a different effect on *g-mean* depending on the magnitude of *acc+*: the smaller the value of *acc+*, the greater the change of g-mean. This property means that the "cost" of misclassifying each positive example increases the more often positive examples are misclassified. A learning system based on g-mean is thereby forced to produce hypotheses that correctly classify a non-negligible fraction of the positive training examples. On the other hand, g-mean is less than ideal for filtering tasks because it ignores precision.

This behavior is depicted in Figure 2 that shows the performance achieved by C4.5 (Quinlan, 1993) for varying numbers of lookalikes while the set of oil spills remains unchanged. The figure shows the detrimental effect that severe imbalance in the class distribution can have on the quality of the classifier: the g-mean and the accuracy on the positives both decrease considerably as the number of negative examples increase. On account of the dominant representation of negative examples, the average accuracy would generally follow the dashed curve, "pretending" improvement in a situation where the utility of the system actually drops.

Small Batches of Training Examples

As already mentioned, the examples were naturally grouped: examples drawn from the same image constitute a single *batch*. Whenever data is collected in batches, there is a possibility that the batches systematically differ from one another, or that there is a much greater similarity of examples within a batch than between batches. In our domain, for example, the ex-

Figure 2: Performance of C4.5 for different numbers of negative examples. Solid: g-mean; dashed: accuracy on negative examples; dotted: accuracy on positive examples.



act parameter settings of the radar imaging system or low-level image processing are necessarily the same for examples within a batch but could be different for different batches. Clearly, in our case, the classifier will be learned from one set of images, and it will be applied on images that were not part of this set. This fact should be taken into account in the evaluation of the system.

This problem has been mentioned by several other authors, including Fawcett and Provost (1996), Ezawa, Singh, and Norton (1996), Kubat, Pfurtscheller, and Flotzinger (1994), and Pfurtscheller, Flotzinger, and Kalcher (1992). For instance, in the SKICAT system (Fayyad, Weir, and Djorgovski, 1993), the "batches" were plates, from which image regions were selected. When the system trained on images from one plate was applied on a new plate, the classification accuracy dropped well below that of manual classification. The solution used in SKICAT was to normalize some of the original features.

One of the chief methodological issues is the requirement that the classifier be trained on one set of images and tested on another set of images. Table 3 illustrates this point using some results obtained from experimenting with SHRINK ($\theta = 0$). The first row (CV) contains the results obtained using the 10-fold cross-validation (average from 5 random runs) applied to the dataset containing all the examples from all images (so that examples from the same image can occur in both the training and the testing sets). These results are clearly superior to those in the second row (LOBO), which are obtained using the leave-one-image-out methodology.

The experiment indicates that the images differ systematically, and therefore cannot be safely combined into one large dataset. This observation suggests another experiment that builds on the following conjecture. In domains where the examples can be mixed, many previous studies have shown that the learning system will give better results if some harmful (e.g. noisy) examples are removed from the training set. This suggests the question: can some of the *batches* be similarly harmful?

Table 3: Leave-one-batch-out (LOBO) and the conventional cross-validation (CV)

	training set			testing set		
	g-mean	acc+	acc-	g-mean	acc+	acc-
CV	74.9	90.6	61.8	70.9	82.5	60.9
LOBO	75.0	85.7	65.6	62.5	78.1	50.1

Table 4: The effect of ignoring one image.

ignored	training set			testing set		
image	g-mean	acc+	acc-	g-mean	acc+	acc-
1	74.7	93.9	59.4	60.0	78.8	45.7
2	71.3	84.9	59.9	61.3	70.3	53.5
3	71.7	84.3	61.1	57.5	79.5	41.6
4	70.0	87.4	56.1	53.3	65.7	43.2
5	75.5	89.0	64.0	62.0	84.6	45.5
6	75.6	94.6	60.5	59.8	81.1	44.1
7	76.5	87.2	67.1	62.6	73.7	53.2
8	81.6	94.4	70.5	67.3	86.1	52.6
9	75.1	87.8	64.2	60.9	64.7	57.2

This can be examined by an experiment where the same leave-one-image-out strategy as in the previous case is applied, only that now one of the images is totally withheld from experimentation: it will appear neither in the training nor in the testing data. The leave-one-image-out strategy was thus applied to 8 out of the 9 images. The i-th row of the table contains the results for the case where the i-th image was ignored. The results achieved by SHRINK (again, $\theta = 0$) are shown in Table 4.

The results in row 4 indicate that the examples contained in image 4 are very typical of the given task, and that removing them from the training set reduces classification performance. The value of g-mean is 53.3% which is much lower than the value achieved when image 4 was not removed (62.5%). Conversely, image 8 seems to be very important (the g-mean, 67.3%, even exceeds the value achieved from *all* images): either its 5 positive examples are very unusual or some of its 80 negative examples are very similar to the positive examples in the other images. The great discrepancies between training and testing set performance in all rows indicate that every image is somewhat different, and also that the training data is insufficient.

Conclusion

The oil spill detection workstation has been delivered, under the name of CEHDS, to Macdonald Dettwiler Associates and will soon undergo field testing in several

European countries. It has image processing suites for two satellites, RADARSAT and ERS-1. Two learning algorithms were included: 1-NN with one-sided selection and SHRINK (Kubat, Holte, and Matwin, 1998). In the latter case, the user can control the rate of false alarms, and trade false alarms for missed oil spills. The user can also decide to retrain the system should more data become available.

In developing the Oil Spill Detection Workstation we faced numerous issues. Most are not specific to the oil spill detection problem: they are the consequence of properties of the application that arise frequently in other machine learning applications. Although each application that has faced these issues has, of necessity, developed some solution, they have not yet been the subject of thorough scientific investigation. They are open research issues of great importance to the applications community.

Perhaps the most important issue is that of imbalanced classes. It arises very often in applications and considerably reduces the performance of standard techniques. Numerous methods for coping with imbalanced classes have been proposed, but they are scattered throughout the literature. At the very least, a large scale comparative study is needed to assess the relative merits of these methods and how they work in combination. Many individual methods, the SHRINK algorithm for example, can undoubtedly be improved by further research.

Learning from batched examples is another issue which requires further research. With the resources (manpower, data) available in this project, we were not able to devise a learning algorithm that could successfully take advantage of the grouping of the training examples into batches. However, we believe further research could yield such an algorithm. Learning from batched examples is related to the issues of learning in the presence of context, as the batches often represent the unknown context in which the training examples were collected. Learning in context has only recently been recognized as an important problem re-occurring in applications of machine learning (Kubat and Widmer, 1996).

Various tradeoffs arose in our project which certainly warrant scientific study. In formulating a problem, one must choose the granularity of the examples (images, regions, or pixels in our application) and the number of classes. Different choices usually lead to different results. For instance, having several classes instead of just two reduces the number of training examples per class but also provides additional information to the induction process. How can one determine the optimal choice? Another tradeoff that arose was between the discriminating power of the features and the number of examples.

In machine learning applications there is no standard measure of performance. Classification accuracy may be useful in some applications, but it is certainly not ideal for all. The research challenge is to develop learning systems that can be easily adapted to different performance measures. For example, cost sensitive learning algorithms work with a parameterized *family* of performance measures. Before running the learning algorithm the user selects a specific measure within this family by supplying values for the parameters (i.e., the costs).

Our experience in this project highlights the fruitful interactions that are possible between machine learning applications and research. The application greatly benefited from—indeed would not have succeeded without—many ideas developed in the research community. Conversely, the application opened new, fertile research directions. Future research in these directions will directly benefit the next generation of applications.

Acknowledgements

The research was sponsored by PRECARN, Inc. The research of the second and third authors is sponsored by the Natural Sciences and Engineering Research Council of Canada, and by Communications and Information Technology Ontario. Thanks are due to Mike Robson, the principal contributor to the image processing subsystem, for his active cooperation with us and for providing invaluable comments on the development of the machine learning subsystem. We also thank Peter Turney (National Research Council of Canada, Ottawa) for his advice on all aspects of this work.

References

Armstrong, R., Freitag, D, Joachims, T., and Mitchell, T. (1998). WebWatcher: A Learning Apprentice for the World Wide Web. In R.S. Michalski, I. Bratko, and M. Kubat, *Machine Learning and Data Mining: Methods and Applications*, Wiley, Chichester

Dietterich, T.G., Hild, H., and Bakiri, G. (1995). A Comparison of ID3 and Backpropagation for English Text-to-Speech Mapping. *Machine Learning*, 18, 51–80

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 13th International Conference on Machine Learning, ICML'96* (pp. 139–147), San Mateo, CA: 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

Fayyad, U.M., Weir, N., and Djorgovski, S. (1993). SKICAT: A Machine Learning System for Automated

Cataloging of Large Scale Sky Surveys. *Proceedings of the 10th International Conference on Machine Learning, ICML'93* (pp. 112–119), San Mateo, CA: Morgan Kaufmann.

Kononenko, I., & 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., Holte, R., and Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*

Kubat, M and Matwin, S. (1997). Addressing the Curse of Imbalanced Training Sets: One-Sided Sampling. *Proceedings of the 14th International Conference on Machine Learning, ICML97*

Kubat, M., Pfurtscheller, G., and Flotzinger D. (1994). AI-Based Approach to Automatic Sleep Classification. *Biological Cybernetics*, 79, 443–448

Kubat, M. and Widmer, G. (1996) (eds.): *Proceedings of the ICML'96 Pre-Conference Workshop on Learning in Context-Sensitive Domains*, Bari, Italy, 1996

Langley, P. and Simon, H. (1998). Fielded Applications of Machine Learning. In R.S. Michalski, I. Bratko, and M. Kubat, *Machine Learning and Data Mining: Methods and Applications*, Wiley, Chichester

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

Lewis, D., & Gale, W. (1994). A Sequential Algorithm for Training Text Classifiers. *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 3–12), Springer-Verlag.

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

Ossen, A., Zamzow, T, Oswald, H., and E. Fleck (1994). Segmentation of Medical Images Using Neural-Network Classifiers. *Proceedings of the International Conference on Neural Networks and Expert Systems in Medicine and Healthcare (NNESMED'94)* (pp. 427–432).

Pfurtscheller, G., Flotzinger, D. and Kalcher, J. (1992). Brain-Computer Interface—A New Communication Device for Handicapped Persons. In W. Zagler (ed.): *Computer for Handicapped Persons: Proceedings of the 3rd International Conference* (pp. 409–415), Vienna

Provost, F. and Fawcett, T. (1997). Analysis and Visualization of Classifier Performance with Nonuniform Class and Cost Distribution. *The 14th International Conference on Machine Learning, ICML'97* (submitted)

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

Riddle, P., Segal, R., and Etzioni, O. (1994). Representation design and brute-force induction in a Boeing manufacturing domain. *Applied Artificial Intelligence*, 8, 125–147

Saitta, L., Giordana, A., & Neri, F. (1995). What Is the "Real World"?. *Working Notes for Applying Machine Learning in Practice: A Workshop at the Twelfth International Conference on Machine Learning*, Technical Report AIC-95-023 (pp. 34–40), NRL, Navy Center for Applied Research in AI, Washington, DC.

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

Turney, P (1995). Data Engineering for the Analysis of Semiconductor Manufacturing Data. *IJCAI-95 workshop on Data Engineering for Inductive Learning*, pp. 50–59.