Outlier Detection by Active Learning

Naoki Abe
IBM T. J. Watson Research Center
1101 Kitchawan Road
Yorktown Heights, NY 10598, USA
nabe@us.ibm.com

Bianca Zadrozny
Instituto de Computação
Universidade Federal Fluminense
Rua Passo da Pátria, 156
Niterói, RJ, Brazil, 24210-240
bianca@ic.uff.br

John Langford*
Toyota Technological Institute at Chicago
1427 E 60th Street
Chicago, IL 60637, USA
jl@tti-c.org

ABSTRACT

Most existing approaches to outlier detection are based on density estimation methods. There are two notable issues with these methods: one is the lack of explanation for outlier flagging decisions, and the other is the relatively high computational requirement. In this paper, we present a novel approach to outlier detection based on classification, in an attempt to address both of these issues. Our approach is based on two key ideas. First, we present a simple reduction of outlier detection to classification, via a procedure that involves applying classification to a labeled data set containing artificially generated examples that play the role of potential outliers. Once the task has been reduced to classification, we then invoke a selective sampling mechanism based on active learning to the reduced classification problem. We empirically evaluate the proposed approach using a number of data sets, and find that our method is superior to other methods based on the same reduction to classification, but using standard classification methods. We also show that it is competitive to the state-of-the-art outlier detection methods in the literature based on density estimation, while significantly improving the computational complexity and explanatory power.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications–*Data Mining*

General Terms: Algorithms, Performance, Design

Keywords: Outlier detection, active learning, ensemble method

1. INTRODUCTION

Outlier detection is an alternative to supervised learning methods, particularly for applications in which label information is either hard to obtain or unreliable. Typical examples of such application areas include network intrusion detection, fraud detection and fault detection in manufacturing, among other things. Most of the existing approaches to the problem of outlier detection in the literature have been based on density estimation methods, and in particular, on nearest-neighbor methods [5, 9, 10, 13]. There are two shortcomings with this type of approach, which we recognize as potential obstacles to real world deployment of these methods. One is that it tends not to provide semantic explanation as to why a particular instance has been flagged as an outlier. The other is the relatively high computational requirement, since nearest-neighbor methods need to store all or a large part of the past examples for effective classification of future examples.

In the present paper, we exhibit a novel approach to outlier detection based on classification, which addresses both of these issues. The approach is based on two key ideas. First, we present a simple reduction of outlier detection to classification, via a procedure that involves applying classification to a labeled data set containing artificially generated examples that play the role of potential outliers, in addition to the actual data. Once the task has been reduced to classification, we can take advantage of the wealth of techniques offered by classification theory. In particular, we invoke the technique of active learning to the reduced classification problem. We do this to address some subtle shortcomings of using the artificial examples approach. In particular, direct application of this approach can fail to work, since the real world examples may adhere to some hidden constraints that the artificial examples violate, and hence it may be trivial to classify the two groups of examples apart. Even though perfect classification may be possible, this would lead to a useless classifier for outlier detection, since it just learned to distinguish the *artificial* examples from the real ones.

More generally, the performance of outlier detection inherently depends on the exact choice of the sampling distribution of the artificial examples. Active learning, by virtue of the way it effectively alters the sampling distribution to focus on the decision boundary between the normal examples and outliers, weakens the dependence on this distribution. Consequently, it also should be free from the shortcomings mentioned above, namely of just picking up the telltale patterns of artificial examples.

Specifically, we employ a selective sampling mechanism based on an active learning framework, which may be termed

*This author's present affiliation is Yahoo Research, 45 W 18th St New York, NY 10011, USA, jl@yahoo-inc.com

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
KDD'06 August 20–23, 2006, Philadelphia, Pennsylvania, USA.
Copyright 2006 ACM 0-59593-339-5/06/0008 ...$5.00.

an *ensemble-based minimum margin approach* (e.g. [2, 11]). The benefits of this approach are two-fold:

1. Selective sampling based on active learning is able to provide improved accuracy for outlier detection, according to the intuition above.

2. The use of selective sampling provides the data scalability that we need for typical applications of outlier detection. That is, it makes it possible to handle very large data sets, possibly in the millions, with light requirements on the computational power and memory.

We empirically evaluate the effectiveness of the proposed approach using a number of data sets that are publicly available. The results demonstrate that our method is superior to other methods based on the reduction to classification but using standard classification methods. In particular, our method outperforms applying bagging and boosting using the same component algorithm on the same reduced problem. Indeed, active learning helps in the present context. The results also indicate that our method is competitive with the state-of-the-art outlier detection methods in the literature, such as the LOF method [5] which has been extensively studied as a method for outlier detection.

2. METHODOLOGY

2.1 Reduction of Unsupervised Learning to Classification

We begin by presenting a non-standard model of unsupervised learning. Assume that the data are drawn from some probability distribution U on an instance space X. The goal in this model is to choose a "good" partition π of the space X. The partition π divides the space X into two subspaces which we call, π and $X - \pi$. A "good" partition π contains "most" of the points while minimizing the size of π. We then define the notion of error for this form of unsupervised learning as follows.

$$e_{U,B}(\pi) = \frac{1}{2}\left(\Pr_{x\sim U}(x \notin \pi) + \Pr_{x\sim B}(x \in \pi)\right)$$

Here we used B to denote the "background" distribution over X, such as the uniform distribution for a finite X. Note that the minimization of the first term attempts to include the set of likely events, while the second term forces exclusion of the unlikely events. Intuitively, the goal here is to find a "small" (w.r.t. B) set π which contains "most" of the data (w.r.t U).

Next, we review a standard model of supervised learning, in particular a *robust* model of classification in which no assumption is made about the target concept. We assume a distribution D over the input space X and the binary output space $Y = \{0, 1\}$. The goal is to find a classifier $h : X \to Y$ with a small true error rate:

$$e_D(h) \equiv \Pr_{x,y\sim D}(h(x) \neq y)$$

We can directly connect these two models, and in so doing transfer much of classification theory to this model of unsupervised learning. In particular, consider the classification problem: decide whether a point is drawn from a distribution B or a distribution U. An element from the distribution $D(x, y)$ is drawn via the following program:

1. Flip a coin with bias 0.5.

2. If "heads"

   (a) then draw $x \sim U$ and return $(x, 1)$

   (b) else draw $x \sim B$ and return $(x, 0)$

Given this distribution, we can interpret any classifier as a partition, and vice versa,

$$c(x) = 1 \Leftrightarrow x \in \pi$$

The error rate of the classifier (in the classifier setting) is intrinsically related to the error rate of the partition in the unlabeled data setting. Now, we can state a meta-theorem connecting true errors in classification to this setting.

PROPOSITION 2.1. *(Translation)*

$$e_D(c_\pi) = e_{U,B}(\pi)$$

where we let c_π denote the classifier corresponding to the partition π, namely: $c_\pi(x) = 1 \Leftrightarrow x \in \pi$

PROOF.

$$e_D(c_\pi) = \Pr_{x,y\sim D}(c_\pi(x) \neq y)$$

$$= \Pr_{x,y\sim D}(c_\pi(x) \neq 0 \text{ and } y = 0) + \Pr_{x,y\sim D}(c_\pi(x) \neq 1 \text{ and } y = 1)$$

$$= \Pr_{x,y\sim D}(c_\pi(x) \neq 0 | y = 0) \Pr_{x,y\sim D}(y = 0)$$

$$+ \Pr_{x,y\sim D}(c_\pi(x) \neq 1 | y = 1) \Pr_{x,y\sim D}(y = 1)$$

$$= 0.5 \Pr_{x\sim B}(x \in \pi) + 0.5 \Pr_{x\sim U}(x \notin \pi)$$

□

The above proposition gives us a direct connection between the error rate of a classifier learned on the distribution D and the error rate of our unsupervised learner. This connection implies a direct reduction since any classifier minimizing the error rate $e_D(c)$ also minimizes the error rate, $e_{U,B}(\pi)$ where π is the partition implied by the classifier.

2.2 Outlier Detection by Active Learning

The proposition may seem simple, but the relationship it establishes is useful in practice, since in some sense it allows us to transfer much of what is known about classification to unsupervised learning, including theory and specific algorithms. In particular, here we apply the idea of active learning, which is a notion that normally makes sense only for supervised learning (classification), to outlier detection.

More specifically, we apply a selective sampling mechanism based on a particular type of active learning methodology, which may be collectively termed *ensemble-based minimum margin active learning*. This approach has its origins in the Query by Bagging procedure due to Abe and Mamitsuka [2], which has been applied and shown effective as a method of selective sampling from a very large data set [11]. Ensemble-based minimum margin active learning combines the ideas of query by committee [14] and ensemble methodology for classification accuracy enhancement [8, 4]. It is given as a sub-procedure a classification algorithm, such as a

decision tree learner like C4.5 or any other classifier of one's choice. It works iteratively, yielding a classifier in each iteration by feeding a sub-sample of the input data set obtained by selective sampling to the given classification learner. The selective sampling in each iteration is dictated by a version of uncertainty sampling, which accepts those examples having lower *margin*, as defined by the ensemble of hypotheses, with higher probability. That is, it scans through the entire data set (or possibly some random subset of it), and performs rejection sampling using an acceptance probability that is calculated as a function of the margin.

Our method, which we call *Active-Outlier*, is based on the idea illustrated above and is presented in Figure 1. Note in the pseudo code that we use $margin(\{\hat{f}_0, ..., \hat{f}_{k-1}\}, x)$ to denote the margin on example x by the ensemble of classifiers, namely the difference between the number of classifiers voting for the most popular label and that for the second most popular label, e.g. for binary classification,

$$margin(\mathcal{F}, x) = |\sum_{\hat{f}\in\mathcal{F}:\hat{f}(x)=1} 1 - \sum_{\hat{f}\in\mathcal{F}:\hat{f}(x)=0} 1|$$

We also let $gauss(\mu, \sigma, Z)$ denote $\int_Z^\infty \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} dx$.

This formulation of sampling probability is based on the following intuition. Consider for simplicity, the binary classification problem. Suppose that n classifiers are chosen at random, each having probability one half of voting one way or another. Then the probability of obtaining k more votes for one label than the other is given by the discrete probability of obtaining $(n+k)/2$ successes in n Bernoulli trials with bias 1/2, and can be approximately calculated using a gaussian distribution. Given that the observed difference is indeed k for a certain example, then by the Bayes theorem, the likelihood of the hypothesis that, for that example, the two labels are equally likely to be predicted by the *rest* of the classifiers, is proportional to the above probability. Assume further that an inclusion of any example is equally likely to contribute to the correct labeling for that example, by the classifier obtained in the current iteration. Since correct prediction by the classifier output in the current iteration will only make a difference if the classifiers in the rest of the iterations are equally split, it should be optimal to include this example with acceptance probability proportional to the above probability.

In the general scheme just described, there are a number of places that admit alternative implementations.

The choice of the underlying distribution B, or how to generate the synthetic sample S_{syn}: This is an obvious point of confrontation, since this choice may be a potentially domain dependent one. In the present paper, we consider two alternative definitions of B: 1) uniform distribution within a bounded sub-space; 2) product distribution of the marginals. Since uniform distribution is not defined on an unbounded domain, we define a bounded subspace by limiting the maximum and minimum to be 10 per cent beyond the observed maximum and minimum, then generate S_{syn} according to the uniform distribution over the bounded domain. For the product of marginals, we estimate the marginal distributions as gaussian distributions for numerical features, and by using observed frequencies for nominal features. For integer features, we then round them off to the nearest integers. Our experimental results are based on the former choice, namely the uniform distribution, except for those on the KDD cup 99 data. We will elaborate on this aspect in the experiments section.

Active-Outlier(Learner A, Samples S_{real}, count t, threshold θ, underlying distrubtion B)

1. Generate a synthetic sample, S_{syn}, of size $|S_{real}|$, according to B.
2. **Let** $S = \{(x,0)|x \in S_{real}\} \cup \{(x,1)|x \in S_{syn}\}$.
3. **For** $i = 1$ **to** t **do**
   (a) Let $M(x) = margin(\{h_0, ..., h_{i-1}\}, x)$.
   (b) Scan through S and obtain S' = **by rejection sampling from** S **with sampling probability:**
   $$gauss(i/2, \sqrt{i}/2, (i + M(x))/2)$$
   (c) **Let** $h_i \equiv A(S')$
   (d) **Let** $\epsilon_i \equiv$ error rate of h_i on S'.
   (e) **Set** $\alpha_i = \log \frac{1-\epsilon_i}{\epsilon_i}$.
4. **Output** $h(x) = \mathbf{sign}\left(\sum_{i=1}^t \alpha_i h_i(x) - \theta(\sum_{i=1}^t \alpha_i)\right)$

Figure 1: Outlier detection method using active learning

The definition of margin: Normally, the margin is defined for classifiers $\hat{f}_i$ outputting $\{0,1\}$ predictions. Here we use a generalization of this to probabilistic classifiers $\hat{f}_i$ outputting conditional probabilities, to take advantage of the finer information provided by the probablities. (See, for example, [12].) More precisely, we define the margin as follows.

$$margin(\mathcal{F}, x) = \sum_{\hat{f}\in\mathcal{F}} \hat{f}(1|x) - \sum_{\hat{f}\in\mathcal{F}} \hat{f}(0|x)$$

We find empirically that this extension yields somewhat better performance.

The choice of ensemble weights α_i: We introduce this weighting to address certain difficulties that multiple authors have noted in the literature about active learning: that inclusion of inaccurate component models into the ensemble can hurt the overall predictive performance. Here we heuristically adopt the weighting scheme of AdaBoost [8]. Experimentally, the weighting has not shown to make a big difference.

Normalizing constant for sampling probability: The rejection sampling formulation for Active-Outlier rests on the assumption that a virtually unlimited stream of examples are available from which to sample. When this assumption is violated, rejection sampling can result in sample sizes that are too small for practical purposes. So we multiply the sampling probability by a normalizing constant $(r/(\sum w_i)$ where w_i is the sampling probability calculated for the i-th example and r is a pre-specified fraction) in each iteration, so that we expect to get roughly the same fraction of examples in each iteration.

2.3 Related Work

The idea of applying classification to outlier detection, using artificially generated examples in place of outliers, is a natural one, and has been employed by multiple authors

in the recent past. Specifically, it has been applied in some empirical studies on anomaly detection tasks in concrete domains such as intrusion detection and image processing [7, 16]. Some theoretical treatment was also provided in our earlier work [1], as well as other work on the more general problem of density-level learning/detection using classification [3]. Recent work by Steinwart et al [15] gives a comprehensive treatment of the classification approach to anomaly detection, based on a theoretical framework of density-level detection. The present work is different from the above body of work because we employ a sampling scheme based on active learning to classification-based outlier detection for the first time and show that this is often crucial for state-of-the-art performance.

3. EMPIRICAL EVALUATION

We empirically evaluate the effectiveness of the proposed outlier detection methodology, using a number of publicly available data sets. We conduct a series of experiments to investigate different questions concerning the performance of the approach for outlier detection using active learning.

In the first set of experiments, we compare the accuracy of the proposed method against other methods that are also based on the reduction to classification, presented in Subsection 2.1. In the second set of experiments, we compare against a well-known outlier detection method based on a modified nearest-neighbor approach, called the LOF method [5]. For the first two comparisons, we mainly use the area under the ROC curve (AUC), as well as the ROC curves themselves as the measure of success.

In the third experimentation, we focus on the specific domain of network intrusion detection, using the well-known data set for the KDD-Cup 1999 network intrusion detection competition. For this setup, we employ the misclassification costs, which were used in the KDD-Cup 1999 competition, as the evaluation criterion, and compare the performance of the proposed scheme against the reported performance in the literature, of a couple representative methods.

3.1 Experimental Procedures

The evaluation of outlier detection methods poses a certain difficulty: there is no clear consensus on what is the definition of an outlier. A natural technical (statistical) definition is to say that a sample is an outlier if and only if it has less likelihood than a certain threshold according to the underlying distribution giving rise to the data. While this is a reasonable definition, in practice it is difficult to use this definition for evaluation. This is because we do not know the underlying distribution, unless the evaluation is done by simulation. For this reason, evaluation is usually done with respect to a more pragmatic notion of outlier detection. That is, one tries to see how well an outlier detection method can be used to separate a rare class of events, often associated with a meaningful notion (such as frauds, intrusions or tool anomalies). Here we take this latter view.

This type of evaluation can be conducted by making use of an existing labeled data set: giving an unlabeled training data set to the outlier detection algorithm and then evaluating the accuracy of flagging examples known to belong to a different class than those included in the training data set as outliers in a test data set. As the outlier class, either a rare class or a class with semantic significance is often chosen. We selected a number of labeled data sets, which have been used in the past for evaluating outlier detection methods. In particular, we chose those data sets used by Lazarevic and Kumar [10] in their empirical evaluation, which have exhibited some evidence they can be reasonably viewed as outlier detection problems. That is, we choose those data sets on which outlier detection methods were found to be able to beat random guessing with some margin. (On some portion of the data sets used in [10], all of the outlier detection methods do no better than random guessing.)

3.2 Comparison with Bagging and Boosting

First we present the results of a comparison between the proposed scheme of *Active-Outlier* using C4.5 as the base classifier learner, against two of the leading classifier learning methods known in the literature, namely Bagging and Boosting, also using C4.5 as the base learner.

The data sets we used for this comparison are Mammography,[1] two versions of Ann-Thyroid, the Shuttle data, and the KDD-Cup 1999 intrusion detection data. We note that Ann-Thyroid and Shuttle are available from the UCI Machine Learning Repository (at http://www.ics.uci.edu/ mlearn/ MLRepository.html) and the KDD-Cup 1999 data set is available from the UCI KDD Archive (http://kdd.ics.uci.edu/). As we explained above, we choose one of the rare classes as the outlier class in our experiments. In deciding which classes should be the outlier classes, and other details about the experimental set-up, we basically follow [10], so that we can directly compare our results (AUC) with those reported in their paper. For example, for experiments with the KDD-cup 1999 data, we only made use of the test data set, both for training and testing by random split as in [10].

The results of this comparison are summarized in Table 1. Some examples of ROC curves for Active-Outlier and Bagging are also exhibited in Figure 2. (We elected not to plot the ROC curves for Boosting, as these were not very informative.) It is seen that Active-Outlier is performing consistently, doing close to best in all cases. Bagging can work very well, although it does quite badly for the KDD-Cup 1999 data. It is quite intriguing that Boosting does not work well at all in this context. Upon a second thought, however, this may not be as surprising as it seems. We are using the classification method to *solve* an artificial classification problem to which we have reduced the original outlier detection problem. This reduced classification problem tends to be highly noisy because the artificial examples are in the background of the real ones. As it is known in the literature, Boosting tends to work poorly in the presence of high noise because it puts too much weight on the incorrectly labeled examples.

3.3 Comparison with LOF and Feature Bagging

In this subsection, we compare the AUC obtained by Active-Outlier with two existing outlier detection methods. The first is the well-established LOF method [9] and the second is the recent Feature Bagging method [10]. The results of this comparison are shown in Table 1. Note that the AUC figures for the two methods have been approximately calculated from ROC graphs reported in [10]. These results show that in all cases, the proposed method achieves AUC that is either roughly equivalent or significantly better than that

[1]The Mammography data set was made available by the courtesy of Aleksandar Lazarevic.

Table 1: Performance of Active-Outlier, Bagging, Boosting, LOF and Feature Bagging: The AUC achieved by each method on each data set is exhibited. The first three methods use J48 (weka version of C4.5).

Data sets	outlier class	Active-Outlier	Bagging	Boosting	LOF	Feature Bagging
Mammography	class 1	**0.81 (± 0.03)**	0.74 (± 0.07)	0.56 (± 0.02)	0.64 (± 0.1)	0.80 (± 0.1)
KDD-Cup 1999	U2R	**0.935 (± 0.04)**	0.611 (± 0.25)	0.510 (0.004)	0.61 (± 0.1)	0.74 (± 0.1)
Shuttle	class 2,3,5,6,7	**0.999 (± 0.0006)**	0.985 (± 0.031)	0.784 (± 0.13)	0.825	0.839
Ann-thyroid	class 1	0.97 (± 0.01)	**0.98 (± 0.01)**	0.64 (± 0.08)	0.869	0.869
Ann-thyroid	class 2	0.89 (± 0.11)	**0.96 (± 0.02)**	0.54 (± 0.01)	0.761	0.769

of Feature Bagging, and outperforms LOF with a significant margin in all cases.

3.4 Comparison on Network Intrusion Detection

The experimental results shown in the earlier sections were with respect to outlier detection rate as summarized by AUC. In this section, we evaluate the competing methods with respect to domain-specific cost information associated with the network intrusion data of KDD-Cup 1999. We use only the data corresponding to the *normal* connections in the training data. More concretely, we use the approximately 200,000 normal connections included in the so-called "10 %" data ("kddcup_data_10_percent.gz"). The synthetic data were generated using the product of the gaussian marginals, rather than the uniform distribution.

We compared the performance of the proposed method against the performance of two other methods reported in the literature, one existing outlier detection method based on density estimation, and one supervised learning method. The outlier detection method we compare against is based on the so-called Parzen Window method, and is documented in [17]. The supervised learning method is the winner of the KDD cup 99 competition (c.f. [6].) We note that the supervised learning method is solving a different problem: (1) it uses labeled data; (2) it is solving a multi-class classification problem with the labels being one of { normal, probe, DOS, U2R, R2L }. Note that the outlier detection methods output only two labels, "normal" and "intrusion". We therefore use a slightly modified cost matrix to evaluate the costs for both types of methods, following [17]. We note that although the costs calculated using the two cost matrices are not identical, the difference is almost negligible. Nonetheless, we use the modified cost matrix to evaluate all of them for a fair comparison.

Table 2 gives the results of this experimentation. What is shown is the total cost incurred on the entire test data, consisting of about 311,000 examples, averaged over 5 randomized runs. The number in parentheses is the standard error. The result for the KDD-Cup 1999 winner is the result of their submitted classifier, hence it is not averaged. No information on the standard error for the Parzen Window method was readily available in the literature.

We note that in getting these results, some trials and errors were made on the choice of the threshold parameter θ in our method. We wish to point out, however, that the excellent performance of this method is not so dependent on the threshold, as is evidenced by the fact that the AUC figures for this method are also very good (c.f. Figure 3.)

The Parzen Window method is a non-parametric density estimation method that basically puts a gaussian distribution around each of the training examples. As such it is an

Method	KDD cup 99 data
ActiveOutlier	59553 (±1460)
Parzen Window	62952
Cup winner	70383

Table 2: Test set costs (standard error) on the KDD-Cup 1999 data obtained by Active-Outlier(40 iterations), Parzen Window, and KDD-Cup 1999 winner.

instance of a nearest neighbor type method, and is considered to be well suited and competitive for outlier detection. Note, however, that it requires the storage of all the training data, and hence its requirement on memory is quite heavy. This is to be contrasted with our proposed method, which by virtue of its iterative sampling approach, is extremely memory light.

It is interesting to see that for this problem, outlier detection methods based on unsupervised learning outperform the best classification method using more information. (The cup winner used all of the 5 million labeled data, whereas both of the anomaly detection methods used (subsets of) the normal connection data included in the so-called 10% data.) This is due, in part, to the unique characteristic of this data set that the test data are drawn from a significantly different data distribution than the training data, which is realistic but makes the problem challenging. Nonetheless, this in some sense gives a strong argument for employing outlier detection based methods for certain real world anomaly detection problems (such as network intrusion and fraud detection).

4. CONCLUSIONS

We have proposed a novel scheme for outlier detection based on the reduction of unsupervised learning to classification, and active learning based selective sampling. In addition to the high detection accuracy verified with our experiments, the proposed approach has the advantage that it is an inherently resource light approach well suited in stream mining set-ups. In addition, the proposed approach yields high explanatory power, and we are finding that this is critical in many real world applications.

5. REFERENCES

[1] N. Abe, C. V. Apte, B. Bhattacharjee, K. A. Goldman, J. Langford, and B. Zadrozny. Sampling approach to resource light data mining. In *Workshop at SIAM 2004 - Workshop on Data Mining in Resource Constrained Environments*, February 2004.

[2] N. Abe and H. Mamitsuka. Query learning strategies using boosting and bagging. In *Proceedings of the*

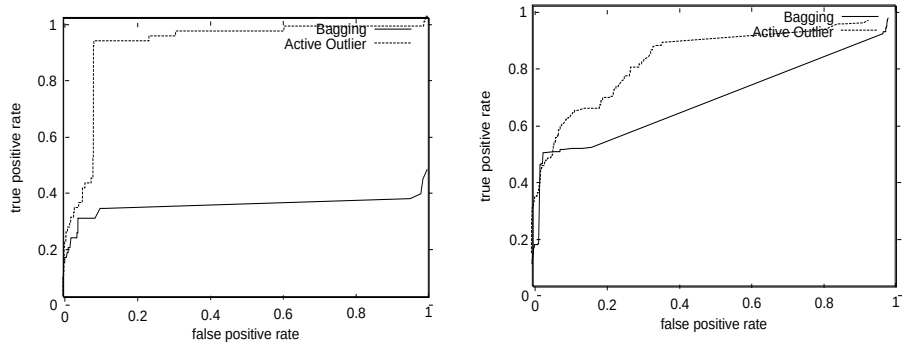

Figure 2: ROC curves for Active Outlier and Bagging for KDD99 and Mammography.

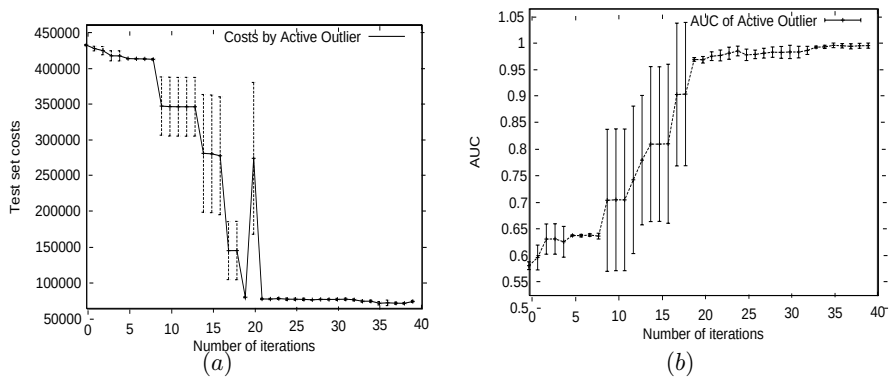

Figure 3: (a) The misclassification cost and (b)AUC by Active Outlier, on the KDD-Cup 1999 data, plotted as a function of the number of iterations, with error bars (standard deviation).

Fifteenth International Conference on Machine Learning, 1998.

[3] S. Ben-David and M. Lindenbaum. Learning distributions by their density levels: a paradigm for learning without a teacher. *Journal of Computer and System Sciences*, 55:171–182, 1997.

[4] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

[5] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander. Identifying density based local outliers. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 2000.

[6] C. Elkan. Results of the kdd'99 classification learning contest. Available at http://www.cs.ucsd.edu/users/elkan/clresults.html, 1999.

[7] W. Fan, M. Miller, S. J. Stolfo, W. Lee, and P. K. Chan. Using artificial anomalies to detect unknown and known network intrusions. In *Proceedings of the First IEEE International Conference on Data Mining (ICDM'01)*, pages 123–130, 2001.

[8] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

[9] E. Knorr and R. Ng. Algorithms for mining distance based outliers in large data sets. In *Proceedings of the Very Large Databases (VLDB) Conference*, August 1998.

[10] A. Lazarevic and V. Kumar. Feature bagging for outlier detection. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2005.

[11] H. Mamitsuka and N. Abe. Efficient mining from large databases by query learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[12] P. Melville and R. Mooney. Diverse ensemble for active learning. In *Proceedings of the 21st International Conference on Machine Learning*, pages 584–591, 2004.

[13] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, May 2000.

[14] H. S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In *Proc. 5th Annu. Workshop on Comput. Learning Theory*, pages 287–294. ACM Press, New York, NY, 1992.

[15] I. Steinwart, D. Hush, and C. Scovel. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6:211–232, 2005.

[16] T. Theiler and D. M. Cai. Resampling approach for anomaly detection in multispectral images. In *Proceedings of the SPIE 5093*, pages 230–240, 2003.

[17] D. Y. Yeung and C. Chow. Parzen-window network intrusion detectors. In *Proceedings of the 16th International Conference on Pattern Recognition (ICPR'02)*, pages 385–388, 2003.