Active Learning for One-Class Classification

Vincent Barnabé-Lortie, Colin Bellinger, Nathalie Japkowicz
School of Electrical Engineering and Computer Science
University of Ottawa
Ottawa, Canada
{vbarn021, cbell052}@uottawa.ca, nat@site.uottawa.ca

***Abstract*—Active learning is a common solution for reducing labeling costs and maximizing the impact of human labeling efforts in binary and multi-class classification settings. However, when we are faced with extreme levels of class imbalance, a situation in which it is not safe to assume that we have a representative sample of the minority class, it has been shown effective to replace the binary classifiers with a one-class classifiers. In such a setting, traditional active learning methods, and many previously proposed in the literature for one-class classifiers, prove to be inappropriate, as they rely on assumptions about the data that no longer stand.**

In this paper, we propose a novel approach to active learning designed for one-class classification. The proposed method does not rely on many of the inappropriate assumptions of its predecessors and leads to more robust classification performance. The gist of this method consists of labeling, in priority, the instances considered to fit the learned class the least by previous iterations of a one-class classification model.

We provide empirical evidence for the merits of the proposed method compared to the available alternatives, and discuss how the method may have an impact in an applied setting.

I. Introduction

In many domains, particularly defense and security domains such as intrusion detection [1] and helicopter gearbox monitoring [2], where data from one class (typically the anomalous one) is rare and cannot be expected to form a coherent body, a common solution is to use a one-class classifier rather than the binary alternative, and to learn only the concept of one of the two classes. One then separates the two classes based on whether or not the one-class learner recognizes their data as belonging to the class it learned. An important advantage of using one-class classifiers instead of binary classifiers in such a situation is that a model can be trained even if no data of the rare class is available.

An important trait of many OCC problems, such as radiation monitoring and disease diagnosis, is that the positive class is the class of critical interest; thus, a fundamental issue in OCC is the induction of a decision boundary that is highly accurate on the positive class. To ensure such accuracy, it can be necessary to select a model that misclassifies negatives at a higher than ideal rate. This is problematic since, in such domains, the detection of a positive instance during application often results in a human analyst reviewing the alarming instance in order to make the final decision on an appropriate course of action. This implies that there is significant motivation for minimizing the number of alarms (false positives) whilst maintaining a high level of recall (true positive rate). This dual goal is particularly difficult to achieve in OCC as a result of the fact that few or, in certain cases, no positive instances are available during training to refine the border. Paradoxically, the inevitable utilization of a human analyst's input by the classifier enables the proposal of a system which, in a manner inspired by active learning, is constantly evolving to lower the false positive rate.

There have been attempts, in the literature, to apply active learning to one-class classifiers. However, they have all relied on assumptions which are often inaccurate in the applications we are considering here. For instance, a common assumption is that it is possible to build a representative sample of the positive class (the one not used to train the classifier), or that a large pool of unlabeled data can be expected to contain instances of that class. These assumptions fail when, like in our applications, one-class classifiers are applied to extremely imbalanced domains where it may be the case that we have no data from the positive class, and cannot safely assume that we will find some amongst unlabeled instanced (Note that we can't assume the opposite either: there *could* be such positive examples, and mislabeling them as negative would be dangerous). In radiation monitoring, for example, our task is to avoid a disaster with no instances of a disaster having previously taken place that can be used to build a representative sample or that would inadvertently be present in a pool of unlabeled data but would have gone unnoticed.

In this paper, we thus propose a new strategy to apply active learning to one-class classifiers, which avoids some of these assumptions. The strategy, inspired by uncertainty sampling [3], consists of selecting the instances from the unlabeled pool which least fit the learned concept according to the current version of the one-class classification model. Since we are dealing with OCC rather than binary classification, uncertainty sampling per se could not be applied and we had to design a method that resembled it in the OCC context. Our experiments suggest that this active learning strategy can lead to significant improvements over state-of-the-art methods for active learning in the OCC context. In addition, this method has the advantage of convenience in many practical scenarios, especially in anomaly detection, as it is usually the case that instances that do not match the concept of "regular" data will be reviewed by an expert anyway.

The remainder of this article is structured as follows: Section II goes over the past work on one-class classification and active learning. Section III provides the details of the

978-1-5090-0287-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICMLA.2015.167

proposed method. Section IV describes the experiments we performed, where the merits of active learning for on one-class classification are assessed versus state of the art alternatives.

II. Related Work

A. One-Class Classification

As previously mentioned, when the goal is to distinguish instances of two classes, but a representative sample can only be obtained for one of these two classes, a common solution is to use one-class classifiers. Hence, some of the strongest ideas from binary classification have been adapted to the one-class setting. For instance, the One-Class ϵSVM [4] learns a hyperplane that separates the concept to be learned from the origin, rather than separate two classes. The goal is that the hyperplane should wrap tightly around the concept and that instances that do not match the concept should be on the origin's side of the hyperplane. Similarly, an unsupervised variant of feedforward neural networks, the autoassociator, can be taught to reconstruct at its output level instances of the available class. The reconstruction error can then be used to distinguish instances that belong to that class from those that do not [2].

Alternatively, one could choose to use methods of outlier detection as one-class learners, where, as above, the detected outliers are classified as not belonging to the learned class. This opens the door to a vast selection of methods. There are those inspired by statistics, such as the one described in Section IV-B1 , where outliers are instances that are considered unlikely by some statistical model (often a Gaussian model; methods based on depth [5], where the "outlierness" of an instance is based on the number of layers (convex hulls) of the dataset one would have to peel off to reach that instance; approaches based on distance, such as the KNN distance approach described in Section IV-B2, or the DB(ε,π) Outliers [6], where an instance is an outlier if less than some percentage π of the other points are closer than some radius ε; and density-based approaches, such as the Local Outlier Factor [7], where an instance is an outlier if the data around it is much sparser than it is around its nearest neighbours.

In comparison to traditional binary classifiers, an important advantage of one-class methods is that they do not require a representative sample of the second class. This is of high importance in some domains, and especially in anomaly detection, because we often do not know what members of the second class will look like. For example, in gamma-ray spectrum anomaly detection, one may know what normal spectra look like, and have plenty of example data, but it is much harder to characterize all possible gamma-ray anomalies, including the ones we have not seen before.

While there have been attempts to leverage available data from the second class to improve a one-class classifier's performance [8], they are left out of the scope of this paper, where we instead investigate what can happen even if no data from that class is available.

B. Active Learning

Active Learning is a form of semi-supervised learning where two separate sets of data are used: a set of labeled instances, used to train the initial model, and to which newly labeled instances will be added; and a set of unlabeled instances, some of which will be selected and labeled by a human expert before being added to the training set for subsequent models.

The argument for active learning is purely economical: We know larger training sets can often help us build a better model. Given costly human resources for labeling and a pool of unlabeled instances, we would like to label as few instances as possible, while getting the best gain in classifier performance. In other words, we want to rank the unlabeled instances so that the instances with higher ranks are likely to be informative instances that will help us refine the decision boundary of our classifier. Obviously, within this framework, the selection procedure is a very important component. *Which examples should we label first?* The literature offers many strategies. For instance:

Lewis and Gale [3] suggest *uncertainty sampling*. The idea is to pick instances for which the current classifier is not very confident in its prediction. Alternatively, we can use *margin sampling* [9], which selects the instances for which the difference in confidence between the top 2 classes is smallest. Those are instances for which two classes seem equally likely, and knowing the true label may help us refine our classifier.

Another popular approach, *Query-By-Committee* [10], maintains a committee of models that each output a prediction for the unlabeled data. We then select for labeling instances for which the members of the committee most disagree. The intuition behind QBC is based on version spaces [11]: by labeling instances in controversial areas of the instance space, QBC effectively reduces the size of the space of hypotheses that match the training data.

Next, there are selection strategies based on expected model change, as introduced in [12]. These select instances for which, regardless of the true label, the expected change in the model when the instance is added to the training set is largest, as measured by some gradient of the model. Similarly, expected error reduction strategies [13] select instances that should lead to a decrease in the error of the model.

Recently, some authors have proposed strategies to apply active learning to one-class classification, and, relatedly, to outlier detection:

Active-Outlier [14] generates artificial outliers before applying traditional active learning methods to the resulting classification problem. However, this approach relies on assumptions about the distribution of outliers, which our method does not require, in order to generate synthetic instances.

In a manner similar to uncertainty sampling [3], the method described in [15] applies active learning to support vector data descriptions by labeling first instances which are located near the surface of the SVDD sphere. Unlike our proposed approach, however, this method is limited to a single algorithm.

In [16], a variation of margin sampling is used to prioritize unlabeled instances for labeling. However, the particular method they use assumes that there is a significant number of instances of the second class in the unlabeled pool, and that differences between the distribution of the unlabeled data and of the labeled normal data are caused by the presence of this class. If data of the second class happen to be absent in the unlabeled pool, Ghasemi et al.'s method breaks apart. Our approach, on the other hand, does not make any such assumption.

Finally, in [17], Ghasemi et al. use kernel density estimation to identify instances in the unlabeled pool that fit the target class best, and label these instances first. Hence, one expects that the instances this method adds to the training set match the current model's idea of the target class. Our method does quite the opposite: we add to the training set instances which challenge our current model's understanding of what fits the learned class, hopefully leading it to refine the concept's border. Our experimental results, as described in section IV-E, appear to give our method the upper hand.

III. Proposed Method

In the context of integrating one-class classification techniques and active learning, we propose a method whose goal is to maximize the performance gains associated with human labeling efforts.

The heart of our method is in the active learning instance selection procedure. In essence, our method's selection strategy is to first pick the instances which the initial model considers to match the learned class the *least*. Obviously, as in all active learning settings, the instances are not then blindly added to the training set: they are sent to a domain expert for labeling. The reasons behind our selection strategy are manifold:

We have to keep in mind that the only instances that will be of use, in a *one-class learning* setup, are the ones that, during labeling, receive the label of the class we are learning. Hence, if our goal is to select informative instances, then we should pick instances that, if they received the label of the learned class, would be informative, or *surprising*, for the current model. Indeed, instances that were previously considered to be very different from the learned class and that turn out to be members of that class are, in a way, surprising for the model.

There is also a practical argument for this selection strategy: in a live environment where experts already look at the instances that are flagged as "different" by our system, those that turn out to be false positives could be used to retrain the model and to avoid repeating the same mistakes. In essence, our selection strategy emulates just that. Therefore, if we find that this selection strategy leads to better performance improvements than random selection, then live systems could be continuously improved by integrating the false positives they produce into the training set.

In another vein, by selecting instances which do not look like the learned class, but which, in reality, do belong to it, we are possibly exploring underrepresented areas of the class. Given the known difficulty that classifiers have learning small disjuncts of a class [18], [19], obtaining more data from these small disjuncts appears to be a reasonable objective towards which our method might contribute.

Besides the selection strategy, the remainder of the active learning framework is unchanged by our method. As it was described in Section II-B, we first learn an initial model, use it to identify instances that do not seem to belong to the learned class, feed the false positives amongst those back into the training set and, finally, train a final, hopefully improved, model. The base classifier should be a one-class learner. The techniques described in Section II-A would all be appropriate choices.

IV. Experiments: Impact of our Method

In this section, we describe the experimental process used to verify whether or not active learning has a positive impact on one-class classifiers. This includes the experimental scenarios that were compared, the base one-class classification methods that were used within the active learning framework and how their parameters were set, the evaluation methodology (including the metrics used), and, finally, the datasets used.

A. Scenarios

We devise an experimental design with 4 scenarios:

- System based on the initial labeled training set
- System based on the training set augmented by randomly choosing instances from the unlabeled examples pool
- System based on the training set augmented through Ghasemi et al.'s method [17], which uses kernel density estimation
- System based on the training set augmented through our active learning selection procedure

Comparison between the latter 3 scenarios and the first will indicate whether an increase in the size of the training set has any benefits with respect to detection performance. To assess the impact of our active learning selection procedure, however, we must compare it to the performance on the randomly augmented dataset. This is to ensure the results we observe are due to the selection procedure's choices rather than simply to the increase in the training set size. We also compare it to Ghasemi et al.'s method [17].

B. One-Class Methods Used

Three different base one-class classification methods are used, and the results for each are analyzed separately.

1) Mahalanobis distance classifier: The Mahalanobis distance [20] is a measure of distance from the mean of a set of instances that differs from the Euclidean distance in that it takes into account correlations in its dataset. Thus, two instances with the same Euclidean distance from the mean of a distribution, if they differ from the mean in different directions, can have very different Mahalanobis distances from that mean.

We flag as belonging to the positive class those instances that have high Mahalanobis distances to our training set of negative instances. This technique has been used in the past [21] on one of the datasets we used (the Saanich data, see section IV-D).

2) Distance to KNN: As an example of distance-based outlier detection, we use the distance to the K$^{\text{th}}$ nearest neighbour of an instance in the training set of negative instances as an "anomaly" score, and flag as belonging to the positive class instances that have high such scores. This method was introduced by Ramaswamy et al in 2000 [22].

This follows the intuition that the negative class is sparser around areas of the positive class, and thus the nearest negative neighbors are further, whereas instances that have multiple nearby negative neighbours are more likely to be negative themselves.

3) One-Class ϵSVM: With the ϵSVM, like with regular support vector machines [23], we learn a separating hyperplane, resting upon a selected set of training instances called the *support vectors*, that serves as our decision boundary. The hyperplane is in a space of higher dimensionality, to which the data is mapped through a kernel function. The distinction between the traditional SVM and the ϵSVM is that, unlike the traditional SVM which use the hyperplane to separate instances of two classes, with the one-class ϵSVM, we are looking for a hyperplane that separates the instances of the class we are learning from the origin of the higher dimensional space.

Hence, at testing time, only instances that fall on the learned class' side of the hyperplane are considered to be of that class.

In our experiments, we use the LibSVM [24] implementation of ϵSVM

C. Evaluation Methodology

Our experiments consist of 10 folds of cross-validation. The assignment of instances to each of the 10 folds is shared across all scenarios, which is important because it allows these iterations to be considered paired trials for the purpose of statistical testing.

Of the 10 folds, we always use 1 for testing, 3 for training the initial model, and 6 as the pool of unlabeled examples for the active learning selection procedure to pick from. The folds rotate so that each fold is used for each role evenly. The active learning selection procedure was allowed to choose up to one sixth of the examples in the unlabeled pool.

For each iteration, we draw ROC curves by using the scores given to each of the testing background instances as thresholds, alternately, and calculating, at each threshold, the number of true and false positives. A curve is drawn for each of the 3 scenarios. In addition to plotting the curves, we calculate the area underneath, which is used as our performance metric for statistical testing purposes.

We also identify the point on the ROC curve closest to the (0, 1) point (as measured by the Euclidean distance). This point is used to calculate a true positive rate and a false positive rate for the classifier. These two metrics offer a more applied perspective of the classifier's performance.

Friedman's non-parametric test is used to assess whether there is a difference in performance between the three scenarios. When there is, Nemenyi's post-hoc test is used to identify the scenarios between which this difference is important.

Dataset	*Initial*	*Random*	*KDE*	*Active*
Pen Digits	0.9409	**0.9415**	0.9413	0.9352
MAGIC	**0.8036**	0.8035	0.8025	0.7969
Saanich	0.7126	0.7122	0.7128	**0.7134**

(a) Mahalanobis Distance

Dataset	*Initial*	*Random*	*KDE*	*Active*
Pen Digits	0.9876	0.9896	0.9872	**0.9935**
MAGIC	0.7708	0.7741	0.7734	**0.7789**
Saanich	0.7208	0.7191	0.7203	**0.7219**

(b) Distance to KNN

Dataset	*Initial*	*Random*	*KDE*	*Active*
Pen Digits	0.9929	0.9937	0.9930	**0.9954**
MAGIC	0.7805	0.7850	0.7839	**0.7898**
Saanich	0.7533	0.7547	0.7557	**0.7587**

(c) One-Class SVM

TABLE I

RESULTS OF THE FIRST EXPERIMENT (MEASUREMENTS ARE THE AUROC AVERAGED OVER 10-FOLD CV)

D. Datasets Used

We test our method on three different datasets. The first two, *Pen Digits* and the *MAGIC Gamma Telescope* datasets, were taken from the UCI Machine Learning Repository [25]. Since these are datasets generally used for multiclass classification, they had to be converted to a binary format: classes were combined into two groups, only one of which was used to train the one-class classifier.

The third dataset was given to us by our collaborators at the Radiation Protection Bureau of Health Canada. It consists of 19113 samples from a Sodium iodide detector in the city of Saanich, British Columbia, collected over a period of 7 months. The sampling period was of 15 minutes. Each sample consists of photon counts over 512 energy bins, but only the first 250 were used following advice from domain experts. The photon counts are non-negative integers. Of the 19113 instances, 95 are anomalies and 19018 are normal. On this dataset, it was not possible to use Ghasemi et al.'s active learning method because of the data's dimensionality, which causes sparsity: when applying kernel density estimation to the Saanich dataset, the probability density around points in the dataset was always so small that our MATLAB environment rounded down to zero.

E. Results

Tables Ia, Ib and Ic present the results obtained with, respectively, the Mahalanobis distance, the KNN distance and the One-Class ϵSVM, on the three domains. The measurements shown are the area under the ROC curve obtained by varying the threshold at which instances are determined to belong to one class or the other. We report the average measurement over 10 folds of cross-validation, for each of the four scenarios: *Initial*, *Random*ly Augmented, Augmented through Ghasemi

	TPR		FPR	
Dataset	Before	After	Before	After
Pen Digits	0.8896	0.8800	0.1447	0.1550
MAGIC	0.7127	0.7107	0.2510	0.2592
Saanich	0.5611	0.5600	0.1586	0.1623

(a) Mahalanobis Distance

	TPR		FPR	
Dataset	Before	After	Before	After
Pen Digits	0.9603	0.9697	0.0470	0.0302
MAGIC	0.6587	0.6555	0.2520	0.2280
Saanich	0.5863	0.5779	0.2016	0.1779

(b) Distance to KNN

	TPR		FPR	
Dataset	Before	After	Before	After
Pen Digits	0.9698	0.9719	0.0370	0.0261
MAGIC	0.6678	0.6811	0.2467	0.2355
Saanich	0.6242	0.6337	0.2254	0.2344

(c) One-Class SVM

TABLE II

RESULTS OF THE FIRST EXPERIMENT (MEASUREMENTS ARE THE TRUE AND FALSE POSITIVE RATES AVERAGED OVER 10-FOLD CV)

et al.'s method based on kernel density estimation (*KDE*) and Augmented by *Active* Learning.

Tables IIa, IIb, and IIc present the true and false positive rates obtained when using the initial training set (before) and our active learning method (after). To calculate these true and false positive rates, the point closest to (0, 1) on the ROC curve was used.

The following subsections give an overview of statistical tests on the AUROC results.

1) Mahalanobis Distance: As we can see from the results, when using the Mahalanobis distance as the base method, the active learning strategy seems to *hurt* performance rather than help it.

This is confirmed to be statistically significant on the first two domains: the probabilities of obtaining such results under the null hypothesis of the Friedman's test are, respectively, 4.155×10^{-4} and 7.4×10^{-3} for the pen digits and MAGIC domains. In both cases, Nemenyi's test indicates that the active learning technique is significantly outperformed by the other techniques.

On the Saanich dataset, Friedman's test finds no significant difference between the four scenarios.

2) KNN Distance: In this case, the active learning technique does seem to lead to better results on average. Its mean performance rank is systematically better than the other two scenarios across all three datasets. On the first two, Friedman's test gives us a p-value of 1.380×10^{-6} and 4.709×10^{-5}. On both, Nemenyi's test indicates that the active learning scenario outperforms the *Initial* and *KDE* scenarios.

On the Saanich dataset, although the trend seems to be the same, with the active learning technique generally coming out on top, the difference is not large enough for statistical significance.

3) One-Class ϵSVM: Finally, with the one-class ϵSVM, we obtain similar results to the KNN distance. Again, active learning seems to lead to the best results.

Once again, on the first two domains, Friedman's test rejects the null hypothesis. The p-values are 4.7×10^{-3} and 2.323×10^{-6}. On both, Nemenyi's test finds that active learning leads to a significant improvement over the *Initial* and *KDE* scenarios. Only on the first domain, Pen Digits, is the advantage of *Active* over *Random* significant according to this test.

F. Discussion

The disparity between the results obtained with the Mahalanobis distance technique and the other two is an interesting phenomenon. Although this would be difficult to prove, we suspect it may be because of their respective parametric and non-parametric natures.

At the core of the Mahalanobis distance classifier is the assumption of a gaussian distribution. The distance is essentially an indicator of how unlikely an instance is under that distribution. The selection procedure we used essentially oversamples the tails of this gaussian distribution, creating multiple additional modes in the data's distribution. Our hypothesis as to why the Mahalanobis distance classifier performed worse with the active learning is that it is unable to properly handle this artificial multimodal distribution. Another way to see it is through the characteristics of parametric classifiers. With a finite number of parameters, the Mahalanobis model has no choice but to *change* its parameters, to alter its representation of the data it is modeling. Essentially, after the active learning procedure is followed, it is given data with an artificially high variance, because of the added anomalies.

On the other hand, non-parametric classifiers are capable of growing in complexity when given more training data. Rather than *changing* their understanding of the data they learned, they can *add onto it* new subconcepts. The addition of new knowledge does not require them to forget part of what they knew before. This is particularly true with support vector machines, in this case: these models select a set of instances close to the decision boundary as support vectors. The instances we add through active learning are likely to become candidates for support vectors in the next model, if they are close to the decision boundary. Hence, the active learning process could be providing the SVM with a better selection of support vectors.

Overall, there is still room for more research towards a better understanding of the impact of active learning on one-class SVMs. The results here give us the general trend, but future work could look into more detail at how the models evolve. As a primer on what might then be observed, we have noticed through our experiments that, in general, the randomly augmented training set leads to SVMs with more support vectors than the original, which is not surprising given the larger variety of instances to choose from, but that, ultimately, it was the training set augmented through active learning that brought about the SVMs with the largest set of support vectors, suggesting that active learning provided the learner with an even better selection of potential support vectors than random augmentation.

In another vein, we compared our method to Ghasemi et al.'s. Our results suggest that, on these particular domains, and with the two non-parametric one-class classifiers we used, our method is the only one of the two which manages to effect significant performance improvements. This may be because the kernel density based method, by selecting instances in dense areas of the majority class, does not actually provide the distance-to-KNN and One-Class SVM classifiers with useful instances, i.e. instances that will either become nearest neighbours on the border of the class, or support vectors. It is worth mentioning, however, that Ghasemi et al.'s did not, like our method, adversely affect the Mahalanobis classifier.

Now if we look at the results for the true and false positive rates, we have a better idea of how these AUROC results may translate into concrete results for users of the resulting system. As expected, the Mahalanobis distance results are disappointing here as well. However, the Distance to KNN results are much more interesting in this perspective: while the true positive rate did not change much, the false positive rate sometimes dropped by more than 2 percentage points, i.e. 12% fewer false positives. Unfortunately, the One-Class SVM results are less clear: while the true positive rate always improved, the false positive rate did not improve on the Saanich domain.

It is this reduction the false positive rate that we believe may have the largest impact in applications of one-class classifiers. A 12% decrease in the rate of false positives is particularly encouraging when experts need to inspect every instance flagged as positive.

V. Conclusion

In this paper, we discussed a method for performing active learning with one-class classifiers. This method, in contrast with what was already available in the literature, does not make any assumptions about the availability of data from the minority class.

Throughout our experiments, we compared the proposed method to random sampling, and to the method by Ghasemi et al. [17] Our results suggest that while our method may not be appropriate with some parametric models such as a classifier based on the Mahalanobis distance, it appears to outperform its competition with the Distance to KNN and One-Class SVM.

In addition, the performance of our method in terms of its impact on the number of true and false positives detected by a system may make it a valuable addition in many applications.

We explored many avenues for further work, such as the link between the number of labeled instances and the performance. We unfortunately could not include these for lack of space.

Acknowledgment

The authors would like to thank Health Canada for their gracious support of this research.

References

[1] D. E. Denning, "An intrusion-detection model," *Software Engineering, IEEE Transactions on*, no. 2, pp. 222–232, 1987.

[2] N. Japkowicz, C. Myers, M. Gluck *et al.*, "A novelty detection approach to classification," in *IJCAI*, 1995, pp. 518–523.

[3] D. D. Lewis and W. A. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*. Springer-Verlag New York, Inc., 1994, pp. 3–12.

[4] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[5] J. W. Tukey, *Exploratory Data Analysis*. Addison-Wesley, 1977.

[6] E. M. Knorr and R. T. Ng, "A unified approach for mining outliers," in *Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 1997, p. 11.

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM Sigmod Record*, vol. 29, no. 2. ACM, 2000, pp. 93–104.

[8] D. M. Tax, "One-class classification; concept-learning in the absence of counter-examples," *ASCI dissertation series*, vol. 65, 2001.

[9] T. Scheffer, C. Decomain, and S. Wrobel, "Active hidden markov models for information extraction," in *Advances in Intelligent Data Analysis*. Springer, 2001, pp. 309–318.

[10] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the fifth annual workshop on Computational learning theory*. ACM, 1992, pp. 287–294.

[11] T. M. Mitchell, "Generalization as search," *Artificial intelligence*, vol. 18, no. 2, pp. 203–226, 1982.

[12] B. Settles, M. Craven, and S. Ray, "Multiple-instance active learning," *Advances in Neural Information Processing Systems (NIPS)*, vol. 20, pp. 1289–1296, 2008b.

[13] N. Roy and A. McCallum, "Toward optimal active learning through monte carlo estimation of error reduction," *ICML, Williamstown*, 2001.

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

[15] N. Görnitz, M. Kloft, and U. Brefeld, "Active and semi-supervised data domain description," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009, pp. 407–422.

[16] A. Ghasemi, H. R. Rabiee, M. Fadaee, M. T. Manzuri, and M. H. Rohban, "Active learning from positive and unlabeled data," in *Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 244–250.

[17] A. Ghasemi, M. T. Manzuri, H. R. Rabiee, M. H. Rohban, and S. Haghiri, "Active one-class learning by kernel density estimation," in *Machine Learning for Signal Processing (MLSP), 2011 IEEE International Workshop on*. IEEE, 2011, pp. 1–6.

[18] G. M. Weiss, "Learning with rare cases and small disjuncts," in *ICML*, 1995, pp. 558–565.

[19] T. Jo and N. Japkowicz, "Class imbalances versus small disjuncts," *ACM SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 40–49, 2004.

[20] P. C. Mahalanobis, "On the generalized distance in statistics," *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[21] S. Sharma, C. Bellinger, N. Japkowicz, R. Berg, and K. Ungar, "Anomaly detection in gamma ray spectra: A machine learning perspective," in *Computational Intelligence for Security and Defence Applications (CISDA), 2012 IEEE Symposium on*. IEEE, 2012, pp. 1–8.

[22] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," in *ACM SIGMOD Record*, vol. 29, no. 2. ACM, 2000, pp. 427–438.

[23] C. Cortes and V. Vapnik, "Support-vector networks," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

[24] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[25] K. Bache and M. Lichman, "UCI machine learning repository," 2013. [Online]. Available: http://archive.ics.uci.edu/ml