Enhancing One-class Support Vector Machines for Unsupervised Anomaly Detection

Mennatallah Amer
Department of Computer Science and Engineering
German University in Cairo, Egypt
mennatallah.amer @student.guc.edu.eg

Markus Goldstein
German Research Center for Artificial Intelligence (DFKI GmbH)
D-67663 Kaiserslautern, Germany
Markus.Goldstein@dfki.de

Slim Abdennadher
Department of Computer Science and Engineering
German University in Cairo, Egypt
slim.abdennadher @guc.edu.eg

ABSTRACT

Support Vector Machines (SVMs) have been one of the most successful machine learning techniques for the past decade. For anomaly detection, also a semi-supervised variant, the one-class SVM, exists. Here, only normal data is required for training before anomalies can be detected. In theory, the one-class SVM could also be used in an unsupervised anomaly detection setup, where no prior training is conducted. Unfortunately, it turns out that a one-class SVM is sensitive to outliers in the data. In this work, we apply two modifications in order to make one-class SVMs more suitable for unsupervised anomaly detection: Robust one-class SVMs and eta one-class SVMs. The key idea of both modifications is, that outliers should contribute less to the decision boundary as normal instances. Experiments performed on datasets from UCI machine learning repository show that our modifications are very promising: Comparing with other standard unsupervised anomaly detection algorithms, the enhanced one-class SVMs are superior on two out of four datasets. In particular, the proposed eta one-class SVM has shown the most promising results.

Keywords

One-Class SVM, Outlier Detection, Outlier Score, Support Vector Machines, Unsupervised Anomaly Detection

1. INTRODUCTION

Anomalies or outliers are instances in a dataset, which deviate from the majority of the data. Anomaly detection is the task of successfully identifying those records within a given dataset. Applications that utilize anomaly detection include intrusion detection [22], medical diagnosis [17], fraud detection [29] and surveillance [3].

In the anomaly detection domain, three different learning setups based on the availability of labels exist [7]: Similar to standard classification tasks, a supervised learning approach can be used to detect anomalies. In this case, a training dataset containing normal and outlying instances, which is used to learn a model. The learned model is then applied on the test dataset in order to classify unlabeled records into normal and anomalous records. The second learning approach is semi-supervised, where the algorithm models the normal records only. Records that do not comply with this model are labeled as outliers in the testing phase. The last learning setup is unsupervised. Here, the data does not contain any labeling information and no separation into a training and testing phase is given. Unsupervised learning algorithms assume that only a small fraction of the data is outlying and that the outliers exhibit a significantly different behavior than the normal records.

In many practical application domains, the unsupervised learning approach is particularly suited when no labeling information is available. Moreover, in some applications the nature of the anomalous records is constantly changing, thus obtaining a training dataset that accurately describe outliers is almost impossible. On the other hand, unsupervised anomaly detection is the most difficult setup since there is no decision boundary to learn and the decision is only based on intrinsic information of the dataset.

Unsupervised anomaly detection algorithms can be categorized according to their basic underlying methodology [7]. The most popular and also often best performing category for unsupervised learning are nearest-neighbor based methods. The strength of those algorithms stem from the fact that they are inherently unsupervised and have an intuitive criteria for detecting outliers. Their limitations include the quadratic computational complexity and a possible incorrectness when handling high dimensional data.

Support Vector Machines are today a very popular machine learning technique that can be used in a variety of applications. This includes for example handwritten digit recognition, object recognition, speaker identification, text categorization [6] and also anomaly detection. In those applications, SVMs perform at least as good as other methods in terms of the generalization error [6]. SVMs take the capacity of the model into account, which is the flexibility of the learned model to represent any training dataset with a minimal error. This makes SVMs a Structure Risk Minimization (SRM) procedure which is a stimulating alternative to the traditional Empirical Risk Minimization (ERM) procedures.

ODD'13, August 11th, 2013, Chicago, IL, USA.
Copyright 2013 ACM 978-1-4503-2335-2 ...$15.00.

There are many factors that contributed to the high popularity of SVMs today. First of all, its theory is heavily investigated and it comes with a convex optimization objective ensuring that the global optimum will be reached. Moreover, its solution is sparse making it really efficient in comparison to other kernel-based approaches [4]. Finally, some kernels even allow SVMs to be considered as a dimensionality reduction technique [32]. Thus it is argued that it can be used to overcome the "curse of dimensionality", which make SVMs theoretically very attractive for the unsupervised anomaly detection problem.

2. RELATED WORK

As already mentioned, the most popular category for unsupervised anomaly detection are nearest-neighbor based algorithms. Here, global methods, for example the k-nearest neighbor [23, 2] and local methods exist. For the latter a huge variety of algorithms have been developed, many based on the Local Outlier Factor (LOF) [5]: the Connectivity-Based Outlier Factor (COF) [27], the Local Outlier Probability (LoOP) [15], the Influenced Outlierness (INFLO) [14] and the parameter-free Local Correlation Integral (LOCI) [21]. All basically assume that outliers lie in sparse neighborhoods and are far away from their nearest-neighbors [7].

Clustering based algorithms cluster the data and measure the distance from each instance to its nearest cluster center. The basic assumption is that outliers are far away from the normal clusters or appear in small clusters [7]. Algorithms include the Cluster-based Outlier Factor (CBLOF) [12] and the Local Density Cluster-based Outlier Factor (LDCOF) [1]. For the unsupervised anomaly detection problem, the nearest-neighbor based algorithms tend to be more capable of accurately identifying outliers [1]. On the other hand, clustering based anomaly detection has theoretically a lower computational effort, such that it could be preferred in cases where large datasets have to be processed.

Among these two often used categories, also others have been investigated: Classification algorithms, statistical approaches, Artificial Neural Networks (ANNs) and Support Vector Machine (SVMs) [7]. The majority of these categories require a labeled training set and hence they are of little applicability in an unsupervised learning setting. The Histogram-based Outlier Score (HBOS) is an unsupervised statistical based approach that was suggested in [10]. It computes a histogram for each feature individually and then the univariate results are combined in order to produce the final score. It is significantly faster than the other unsupervised anomaly detection algorithms at the expense of precision. Replicator Neural Networks (RNNs) [11] are a semi-supervised neural network based approach. Here, an artificial neural network is trained such that the output is a replica of the input. The reconstruction error is then used as an anomaly score. Another semi-supervised approach is the one-class SVM [25], a special variant of a SVM that is used for novelty detection. Details of which are covered in Section 3. However, a one-class SVM could also be used in an unsupervised setup. Then, training and testing is applied on the same data. Unfortunately, the training on a dataset already containing anomalies does not result in a good model. This is due to the fact that outliers can influence the decision boundary of a one-class SVM significantly.

In a supervised anomaly detection setting, Mukkamala et al. [20] showed that SVM based algorithms are superior compared to ANN based algorithms for the intrusion detection problem. SVMs had a shorter training time and produced better accuracy. The authors stated that the main limitation of SVMs is the fact that it is a binary classifier only. This limits the breadth of information that can be obtained about the type and degree of intrusions.

One class classification (OCC) is the task of learning to describe a target class in order to effectively identify its members. Following Vapnik's [31] intuition, most approaches attempt to find a boundary around the dataset. The formulation of one-class SVM proposed by Schölkopf et al [25] finds the boundary in the form a hyperplane. This is the formulation that we attempt to enhance. Support vector domain description (SVDD) proposed by Tax et al. [28] strives to find the minimum enclosing hypersphere that best describes the data. Both of the above mentioned formulations produce an equivalent solution in case of constant kernel diagonal entries [25]. Quarter-sphere support vector machines [16] were designed to handle intrusion detection data which have one-sided features centered around the origin. It fixes the center of the quarter sphere at the origin yielding a much simpler linear programming optimization objective. Liu and Zheng [18] proposed a SVM variant called MEMEM that combines between the discriminative capabilities of SVMs and the descriptive capabilities of one-class SVMs. This makes it particularly suited for handling unbalanced datasets. However it is a completely supervised approach.

3. ONE-CLASS SVMs

3.1 Motivation

In contrast to traditional SVMs, one-class SVMs attempt to learn a decision boundary that achieves the maximum separation between the points and the origin [24]. Interestingly this was the initial idea from which traditional supervised SVMs emerged. Its origin date back to the earliest work of Vapnik et al. in 1963 [30]. The idea was hindered by the inability to learn non-linear decision boundaries as well as the inability to account for outliers. Both of these problems were solved by the introduction of kernels and the incorporation of soft margins. A one-class SVM uses an implicit transformation function $\phi(\cdot)$ defined by the kernel to project the data into a higher dimensional space. The algorithm then learns the decision boundary (a hyperplane) that separates the majority of the data from the origin. Only a small fraction of data points are allowed to lie on the other side of the decision boundary: Those data points are considered as outliers.

The Gaussian kernel in particular guarantees the existence of such a decision boundary [24]. By observing that all the kernel entries are non-negative, it can be concluded that all the data in the kernel space lies in the same quadrant. This makes the Gaussian kernel well suited to deal with any arbitrary dataset. Let the function $g(\cdot)$ be defined as follows:

$$g(x) = w^T \phi(x) - \rho \tag{1}$$

where w is the vector perpendicular to the decision boundary and ρ is the bias term. Then, Equation 2 shows the decision function that one-class SVMs use in order to identify normal points. The function returns a positive value for normal points, negative otherwise:

$$f(x) = sgn(g(x)). \tag{2}$$

One-class SVMs are traditionally used in a semi-supervised setting. The output of the algorithm is a binary label specifying whether the point is normal or not.

3.2 Objective

Equation 3 shows the primary objective of one-class SVMs:

$$\begin{gathered} \min_{w,\xi,\rho} \frac{\|w\|^2}{2} - \rho + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i \\ \text{subject to: } w^T\phi(x_i) \geq \rho - \xi_i,\, \xi_i \geq 0, \end{gathered} \tag{3}$$

where ξ_i is the slack variable for point i that allows it to lie on the other side of the decision boundary, n is the size of the training dataset and ν is the regularization parameter.

The deduction from the theoretical to the mathematical objective can be stated by the distance to the decision boundary. The decision boundary is defined as:

$$g(x) = 0. \tag{4}$$

In this context, the distance of any arbitrary data point to the decision boundary can be computed as:

$$d(x) = \frac{|g(x)|}{\|w\|}. \tag{5}$$

Thus, the distance that the algorithm attempts to maximize can be obtained by plugging the origin into the equation yielding $\frac{\rho}{\|w\|}$. This can also be stated as the minimization of $\frac{\|w\|^2}{2} - \rho$.

The second part of the primary objective is the minimization of the slack variables ξ_i for all points. ν is the regularization parameter and it represents an upper bound on the fraction of outliers and a lower bound on the number of support vectors. Varying ν controls the trade-off between ξ and ρ.

To this end, the primary objective is transformed into a dual objective, shown in Equation 6. The transformation allows SVMs to utilize the kernel trick as well as to reduce the number of variables to one vector. It basically yields a Quadratic Programming (QP) optimization objective.

$$\begin{gathered} \min_{\alpha} \frac{\alpha^T Q\alpha}{2} \\ \text{subject to: } 0 \leq \alpha_i \leq \frac{1}{\nu n}, \sum_{i=1}^{n}\alpha_i = 1, \end{gathered} \tag{6}$$

where Q is the kernel matrix and α are the Lagrange multipliers.

3.3 Outlier Score

A continuous outlier score reveals more information than a simple binary label such as the output of Equation 2. Similar to [1], our goal is to compute an anomaly score such that a larger score corresponds to significantly outlying points.

In Equation 7, we propose a possible way to compute such a score. Here, g_{max} is the maximum directed distance between the dataset points and the decision boundary. The score is scaled by that distance such that the points that are lying on the decision boundary would have an outlier score of 1.0 similar to [5]. A score larger than 1.0 indicates that the point is a potential outlier.

$$f(x) = \frac{g_{max} - g(x)}{g_{max}} \tag{7}$$

3.4 Influence of Outliers

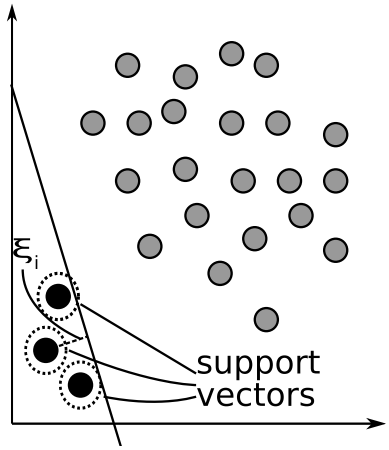


Figure 1: A 2 dimensional example of the decision boundary in the kernel space learned by a one-class SVM.

Figure 1 shows an example of a resulting decision boundary in the presence of outliers in the dataset. The decision boundary is shifted towards the outlying black points and they are additionally the support vectors in this example. Thus the outliers are the main contributors to the shape of the decision boundary. Whilst the shifting of the decision boundary might not have a great influence on the overall rank of the points when using Equation 7, the shape of the decision boundary will. To overcome this problem, in the following section two methods are proposed in order to make the decision boundary less dependent on these outliers.

4. ENHANCING ONE-CLASS SVMs

In this section two approaches are proposed to tackle the challenge that outliers do significantly contribute to the decision boundary. Both approaches are inspired from work done in order to make traditional supervised SVMs more robust against noise in the training dataset. They have the additional advantage of maintaining the sparsity of the SVM solution.

4.1 Robust One-class SVMs

4.1.1 Motivation

The approach is based on Song et al. [26], where the authors attempted to make the supervised SVM more robust in case of existing outliers in the training data. The key idea is the minimization of the of the Mean Square Error (MSE) for tackling outliers using the center of class as an averaged information. The conducted experiments showed that the generalization performance improved and the number of support vector decreased compared to the traditional SVM.

The main modification of robust one-class SVMs is with respect to the slack variables. As illustrated in Figure 1, a non-zero slack variable ξ_i allows a point x_i to lie on the other side of the decision boundary. In the case of robust one-class SVMs, the slack variables are proportional to the distance to the centroid. This allows points that are distant from the center to have a large slack variable. Since the slack variables are fixed, they are dropped from the minimization objective. On the one hand, this causes the decision boundary to be shifted towards the normal points. On the other hand, it loses part of the interpretability of the results as there is no restriction on the number of points that can appear on the other side of the decision boundary. Theoretically, all the points can be labeled as outlying using Equation 2 and consequentially, the majority could have a score greater than 1.0 when using Equation 7.

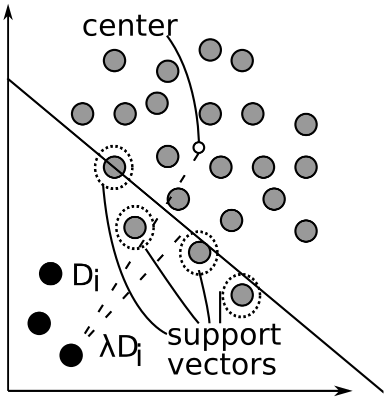


Figure 2: Modifying the slack variables for robust one-class SVMs. Each slack variable is proportional to the distance to the centroid. Dropping the minimization of the slack variables from the objective function causes the decision boundary to be shifted towards the normal points.

Figure 2 illustrates how the slack variables are modified. Points that are further away from the center of the data are allowed to have a larger slack variable. Then, the decision boundary is shifted towards the normal points and the outliers are no longer support vectors.

4.1.2 *Objective*

The objective of the proposed robust one-class SVMs is stated in Equation 8. Here, the slack variables are dropped from the minimization objective. They only appear in the constraints as $\hat{D_i}$, whereas λ is the regularization parameter.

$$\begin{gathered}\min{}_{w,\rho}\ \frac{\|w\|^2}{2} - \rho \\ \text{subject to } w^T\phi(x_i) \geq \rho - \lambda * \hat{D_i}\end{gathered} \tag{8}$$

The slack variable D_i is computed using Equation 9. It represents the distance to the centroid in the kernel space. Since the transformation function is implicitly defined by the kernel (Q), Equation 9 can not directly be used. Thus, an approximation that was introduced by Hu et al [13] is computed instead. This approximation is summarized in Equation 10. Here, the expression $\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$ is a constant and hence it can be dropped. The normalized distance $\hat{D_i}$ appears in the optimization Objective 8.

$$\begin{gathered}D_i = \|\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\|^2 \\ \hat{D_i} = \frac{D_i}{D_{max}}\end{gathered} \tag{9}$$

$$\begin{aligned}D_i &= \|\phi(x_i) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\|^2 \\ &= Q(x_i,x_i) - \frac{2}{n}\sum_{j=1}^{n}Q(x_i,x_j) - \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)\frac{1}{n}\sum_{i=1}^{n}\phi(x_i) \\ &\approx Q(x_i,x_i) - \frac{2}{n}\sum_{j=1}^{n}Q(x_i,x_j)\end{aligned} \tag{10}$$

The dual objective of the robust one-class SVM can be summarized as follows:

$$\begin{gathered}\min{}_{\alpha}\ \frac{\alpha^T Q\alpha}{2} + \lambda D^T\alpha \\ \text{subject to } 0 \leq \alpha \leq 1, e^T\alpha = 1\end{gathered} \tag{11}$$

It can be seen that it is only a minor modification to the dual objective of the one-class SVM objective in Equation 6 and hence it can be incorporated easily in the original solver.

4.2 Eta One-class SVMs

4.2.1 *Motivation*

In contrast to robust one-class SVMs, this approach uses an explicit outlier suppression mechanism. The methodology for supervised SVMs was first proposed by Xu et al. [33]. This suppression mechanism is achieved by introducing a variable η, which represents an estimate that a point is normal. Thus an outlying point would ideally have η set to zero. This variable controls the portion of the slack variables that is going to contribute to the minimization objective.

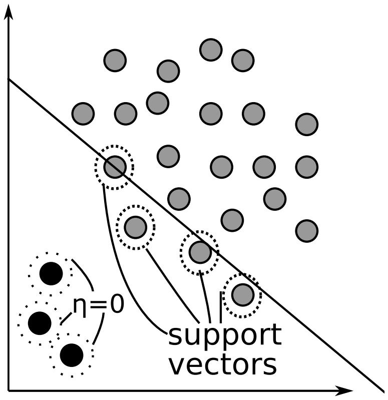


Figure 3: The idea of the eta one-class SVM: Outliers have small values for η and do thus not contribute to the decision boundary.

Figure 3 shows how the introduction of η affects the decision boundary. The outlying points would be assigned $\eta = 0$ thus they would not be considered whilst learning the decision boundary. Here, the decision boundary would be influenced only by the normal points.

4.2.2 *Objective*

Equation 12 shows the objective of the eta one-class SVM. Outlying points would have η set to 0 and hence they would not be contributing to the optimization objective. The disadvantage of introducing η is that the objective loses part of its intuitive interpretation: Minimizing the slack variable is equivalent to minimizing the number of outliers. A variable β is introduced in order to cope with this challenge. It controls the maximum number of points that are allowed to be outlying:

$$\min{}_{w,\rho} \min{}_{\eta_i \in \{0,1\}} \frac{\|w\|^2}{2} - \rho + \sum_{i=1}^{n} \eta_i max(0, \rho - w^T * \phi(x_i)),$$
$$\text{subject to } e^T \eta \geq \beta n. \tag{12}$$

The objective is composed of two parts: A convex quadratic problem in w for a fixed η, and a linear problem in η for a fixed w. However, the objective is not jointly convex. This means that minimizing each part alternatively is not guaranteed to yield a global minimum. The above formulation will be relaxed similar to what was proposed in the original work [33] into a semi-definite problem. Then, it will be relaxed into a iterative formulation due to the limited practicability of semi-definite programs. The iterative relaxation is achieved using concave duality similar to what was used by Zhou et al. [36].

Semi-Definite Programming Problem

The non-convex optimization objective of Equation 12 can be relaxed by relaxing the constraints on η. For a fixed η, introducing Lagrange multipliers would yield the following dual objective:

$$\begin{gathered}\min{}_{0 \leq \eta \leq 1, M = \eta * \eta^T} \max{}_{0 \leq \alpha \leq 1} \frac{\alpha^T Q \cdot M \alpha}{2}, \\ \text{subject to } e^T * \eta \geq \beta n,\ \alpha^T \eta = 1,\ 0 \leq \alpha \leq 1.\end{gathered} \tag{13}$$

The formulation in Equation 13 is convex in both, η and α. The final obstacle is the constraint on matrix M as it is a non-convex quadratic constraint. The constraint can be approximated to $M \succeq \eta * \eta^T$ yielding a convex optimization objective:

$$\min{}_{0 \leq \eta \leq 1} \min{}_{M \succeq \eta * \eta^T} \max{}_{0 \leq \alpha \leq 1} \frac{\alpha^T Q \cdot M \alpha}{2} \tag{14}$$

Objective 14 is equivalent to solving the following semidefinite programming (SDP) problem:

$$\begin{gathered}\min{}_{\eta,\delta,\gamma,\sigma,M} \delta \\ \text{subject to } e^T \eta \geq \beta n, 0 \leq \eta \leq 1, \gamma \geq 0, \sigma \geq 0, \\ M \succeq \eta * \eta^T \\ \begin{bmatrix} 2*(\delta - e^T * \sigma) & (\gamma - \sigma)^T \\ \gamma - \sigma & Q \cdot M \end{bmatrix} \succeq 0 \\ \begin{bmatrix} 1 & \eta^T \\ \gamma - \sigma & Q \cdot M \end{bmatrix} = 0.\end{gathered} \tag{15}$$

Iterative Relaxation

The SDP solution is expensive to compute and hence an alternative approach was proposed by Zhou et al. [36]. It uses a concave duality in order to relax Equation 12 into a multistage iterative problem. A discussion of why the procedure yields a good approximation is given by Zhang [35]. The relaxation yields an objective that has a convex and a concave part, which makes the iterative approach a generalization of a concave convex procedure (OCCC) [34] that is guaranteed to converge.

Let the non-convex regularization in Equation 12 correspond to $g(h(w))$, where $h(w) = max(0, \rho - w^T \phi(x))$ and $g(u) = \inf_{\eta \in \{0,1\}}[\eta^T u]$, using concave duality, the objective can be reformulated into

$$\begin{gathered}\min{}_{w,\rho,\eta} E_{vex} + E_{cave} \\ E_{vex} = \frac{\|w\|^2}{2} - \rho + \eta^T h(w),\ E_{cave} = g^*(\eta),\end{gathered} \tag{16}$$

where g^* is the concave dual of g.

Equation 16 can be solved by iteratively minimizing E_{vex} and E_{cave}. Initially η is set to a vector of ones. Then the following steps are done until convergence:

1. For a fixed η, minimize E_{vex} which corresponds to the following dual objective:

$$\begin{gathered}\min{}_{\alpha} \frac{\alpha^T Q \cdot N \alpha}{2}, \\ \text{where } N = \eta * \eta^T, \\ \text{subject to } \alpha^T \eta = 1,\ 0 \leq \alpha \leq 1.\end{gathered}$$

2. For fixed w and ρ, the minimum of E_{cave} is at:

$$\begin{gathered}u_i = max(0, \rho - w^T \phi(x_i)), \\ \eta_i = I(\beta n - s(i))\end{gathered}$$

   where $s(i)$ is the order of function over u arranged in ascending order and I is the indicator function.

5. EXPERIMENTS

In this section, all the proposed one-class SVM based algorithms are compared against standard nearest-neighbor, clustering and statistical based unsupervised anomaly detection algorithms. The experiments were conducted using RapidMiner [19], where all of the algorithms are implemented in the Anomaly Detection extension[1]. The SVM based algorithms are all using the Gaussian kernel, the spread of the kernel was tuned similar to what is proposed by Evangelista et al. [8]. For the Gaussian Kernel, it is desirable to attain diverse kernel entries as it is a measure of similarity between data points. Evangelista et al. achieved that by maximizing the ratio of the standard deviation of the non-diagonal entries of the kernel matrix to its mean. The maximization objective is solved using gradient ascent.

The area under the ROC curve (AUC) is used as a performance measure, where the curve is created by varying the outlier threshold. It basically measures the quality of the ranking of outliers among normal records. The results of the AUC of running the different algorithms are included in Table 4. Figure 4 shows exemplary ROC curves of the algorithms for two different datasets. In each subfigure, the three SVM based algorithms are compared against the best performing algorithm from each of the other categories.

Another important comparison is between the standard semi-supervised one-class SVM and the proposed improvements of this work: the robust one-class SVM and eta one-class SVM. In addition to the performance measured by the

[1] Available at `http://code.google.com/p/rapidminer-anomalydetection/`

AUC, also the number of support vectors is an important factor to consider as it directly affects the computation time of the SVM based algorithms. The number of support vectors are shown in Table 2. The average CPU execution time of the algorithms over 10 runs is shown in Table 3.

5.1 Datasets

Datasets from the UCI machine learning repository [9] are used for the evaluation of the anomaly detection algorithms. Most of the datasets of UCI repository are traditionally dedicated for classification tasks. Hence they have to be preprocessed in order to serve for the evaluation of unsupervised anomaly detection algorithms. This is typically performed by picking a meaningful outlying class and sampling the outliers to a small fraction [1]. Table 1 summarizes the characteristics of the preprocessed datasets.

The preprocessing was also performed using RapidMiner. For *ionosphere*, *shuttle* and *satellite*, stratified sampling was used to reduce the number of outliers (for reproducibility, the pseudo random generator seed was set to 1992). The preprocessing of the *breast-cancer* dataset was identical to the one proposed in [15].

5.2 Results

The results of the *shuttle* dataset are shown in Figure 4(a). Here, the eta one-class SVM is superior to all the other algorithms. The statistical based algorithm Histogram outperforms the nearest-neighbor and clustering based algorithms. It also outperforms the robust one-class SVM. Surprisingly, the standard one-class SVM outperforms the robust one-class SVM for the shuttle dataset. However, robust one-class produces a much sparser solution with only 5 support vectors dropping the CPU execution by two thirds. Figure 4(b) illustrates the results of the *satellite* dataset. Here, the SVM based algorithms performed worst among all existing categories. The performance of the algorithms is comparable at the first portion of the dataset. Which means that they perform equally well in predicting the top outliers.

Table 4 summarizes the results in terms of AUC for all algorithms on all four datasets. It can be seen that all SVM based algorithms perform generally well on all datasets. For *ionosphere* and *shuttle* the eta one-class SVM is even superior. For the *breast-cancer* dataset, SVM based algorithms score on average. For the satellite dataset, where also many support vectors have been found, results are below the average.

Table 2: Number of support vectors of SVM based algorithms

Algorithm	*ionosphere*	*shuttle*	*breast-cancer*	*satellite*
One-class	106	21374	144	2085
Robust One-class	116	5	90	385
Eta One-class	37	8	48	158

6. DISCUSSION AND CONCLUSION

The experiments showed that the proposed SVM based algorithms are well suited for the unsupervised anomaly detection problem. In two out of four datasets, SVM based algorithms are even superior. They constantly outperform all clustering based algorithms. In general, they perform at least average on unsupervised anomaly detection problems. For the *satellite* dataset, the performance of the SVM based algorithms is slightly below the average. The main reason why this is a challenging dataset for SVM based algorithms is not known exactly, but we can observe that in this case the number of support vectors is comparably high.

When comparing the SVM based algorithms with each other, the eta one-class SVM seems to be the most promising one. On average, it produces a sparse solution and it also performs best in terms of AUC. In general, the robust one-class SVM produces a sparser solution than the standard one-class SVM, but in term of performance, there is no significant improvement. In terms of time efficiency, for larger datasets the enhanced algorithms are more efficient due to the sparsity property.

When looking at computational effort, SVM based algorithms have in general less than a quadratic time complexity due to the sparsity property. However, the parameter tuning for the Gaussian kernel similar to [8] pushes the complexity back to quadratic time.

Additionally, we introduced a method for calculating an outlier score based on the distance to the decision boundary. In contrast to the binary label assigned by standard one-class SVMs, it allows to rank the outliers, which is often essential in an unsupervised anomaly detection setup. The score also has a reference point, which means that scores in the range of [0,1] can be considered to be normal.

In conclusion, SVM based algorithms have shown that they can perform reasonably well for unsupervised anomaly detection. Especially the eta one-class SVM is a suitable candidate for investigation when applying unsupervised anomaly detection in practice.

Acknowledgment

This work was partially funded by the Rheinland-Palatinate Foundation for Innovation, project AnDruDok (961-38 6261 / 1039).

7. REFERENCES

[1] Mennatallah Amer and Markus Goldstein. Nearest-neighbor and clustering based anomaly detection algorithms for rapidminer. *Proc. of the 3rd RapidMiner Community Meeting and Conference (RCOMM 2012)*, pages 1–12, 2012.

[2] Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Principles of Data Mining and Knowledge Discovery*, volume 2431 of *Lecture Notes in Computer Science*, pages 43–78. Springer, 2002.

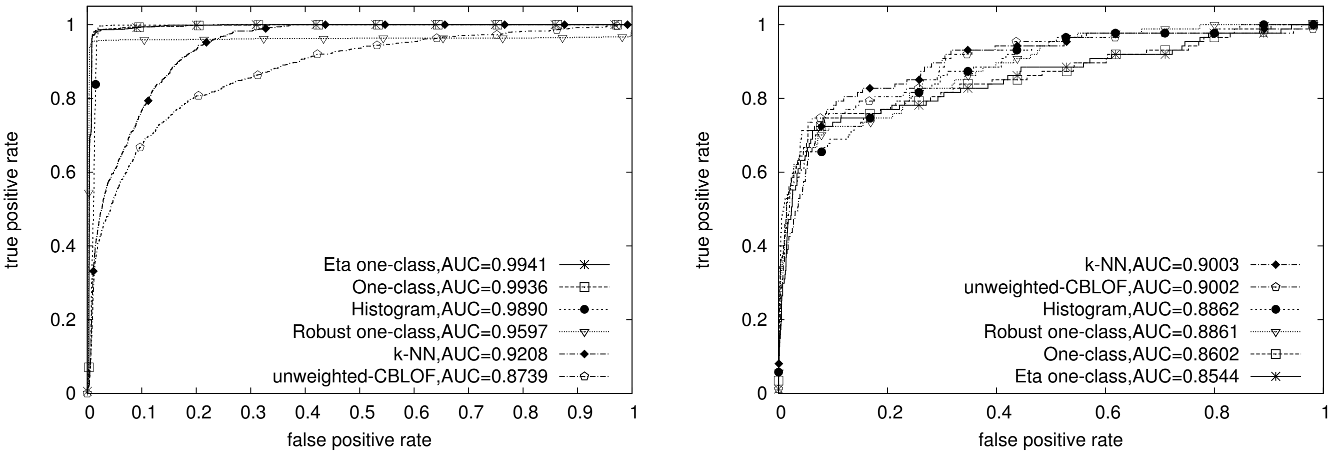
[3] Arslan Basharat, Alexei Gritai, and Mubarak Shah. Learning object motion patterns for anomaly detection and improved object detection. *CVPR*, pages 1–8, 01 2008.

[4] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 1st ed. 2006. corr. 2nd printing 2011 edition, October 2007.

[5] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. In *Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data*, pages 93–104, Dallas, Texas, USA, 05 2000. ACM.

Table 1: Datasets used for evaluation. The preprocessing selects particular classes as outliers and samples it down to a small fraction in order to meet the requirements for unsupervised anomaly detection.

Meta-data	Original Size	Attributes	Outlier class(es)	Resulting dataset size	Sampled outliers percentage
ionosphere	351	26	b	233	3.4%
shuttle	58000	9	2 ,3 ,5 and 6	46464	1.89%
breast-cancer	569	30	M	367	2.72%
satellite	6435	36	2,4 and 5	4486	1.94%



(a) Shuttle

(b) Satellite

Figure 4: ROC curves for SVM based algorithms and existing approaches. For the latter, the best performing algorithms of the categories nearest-neighbor, statistical and clustering based are plotted.

Table 3: CPU execution time of SVM based algorithms

Algorithm	*ionosphere* [ms]	*shuttle* [s]	*breast-cancer* [ms]	*satellite* [s]
One-class	21.55 ± 0.26	747.15 ± 10.94	48.72 ± 1.01	14.02 ± 2.00
Robust one-class	33.82 ± 0.26	218.93 ± 3.17	57.27 ± 2.29	8.60 ± 0.06
Eta one-class	27.48 ± 0.25	4.07 ± 0.14	82.46 ± 0.42	12.35 ± 0.95

Table 4: Comparing the AUC of SVM based algorithms against other anomaly detection algorithms

Dataset	One-class	Robust one-class	Eta one-class	*k*-NN	LOF	COF	INFLO	LoOP	Histogram	CBLOF	u-CBLOF	LDCOF
ionosphere	0.9878	0.9956	**0.9972**	0.9933	0.9178	0.9406	0.9406	0.9211	0.7489	0.3183	0.9822	0.9306
shuttle	0.9936	0.9597	**0.9941**	0.9208	0.6072	0.5612	0.5303	0.5655	0.9889	0.8700	0.8739	0.5312
breast-cancer	0.9843	0.9734	0.9833	0.9826	0.9916	0.9888	**0.9922**	0.9882	0.9829	0.8389	0.9743	0.9804
satellite	0.8602	0.8861	0.8544	**0.9003**	0.8964	0.8708	0.8592	0.8664	0.8862	0.4105	0.9002	0.8657

[6] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):15:1–15:58, July 2009.

[8] Paul F. Evangelista, Mark J. Embrechts, and Boleslaw K. Szymanski. Some properties of the gaussian kernel for one class learning. In *Proc. of the 17th Int. Conf. on Artificial neural networks*, ICANN'07, pages 269–278. Springer, 2007.

[9] A. Frank and A. Asuncion. UCI machine learning repository, 2010.

[10] Markus Goldstein and Andreas Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. In Stefan Wölfl, editor, *KI-2012: Poster and Demo Track*, pages 59–63. Online, 9 2012.

[11] Simon Hawkins, Hongxing He, Graham Williams, and Rohan Baxter. Outlier detection using replicator neural networks. In *In Proc. of the Fifth Int. Conf. and Data Warehousing and Knowledge Discovery (DaWaK02*, pages 170–180, 2002.

[12] Zengyou He, Xiaofei Xu, and Shengchun Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, 2003.

[13] W.J. Hu and Q. Song. An accelerated decomposition algorithm for robust support vector machines. *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 51(5):234–240, 2004.

[14] Wen Jin, Anthony Tung, Jiawei Han, and Wei Wang. Ranking outliers using symmetric neighborhood relationship. In Wee-Keong Ng and et al., editors,

Advances in Knowledge Discovery and Data Mining, volume 3918 of *Lecture Notes in Computer Science*, pages 577–593. Springer, 2006.

[15] Hans-Peter Kriegel, Peer Kröger, Erich Schubert, and Arthur Zimek. Loop: local outlier probabilities. In *Proceeding of the 18th ACM Conf. on Information and knowledge management*, CIKM '09, pages 1649–1652, New York, NY, USA, 2009. ACM.

[16] Pavel Laskov, Christin Schäfer, Igor V. Kotenko, and Klaus-Robert Müller. Intrusion detection in unlabeled data with quarter-sphere support vector machines. *Praxis der Informationsverarbeitung und Kommunikation*, 27(4):228–236, 2007.

[17] Jessica Lin, Eamonn Keogh, Ada Fu, and Helga Van Herle. Approximations to magic: Finding unusual medical time series. In *In 18th IEEE Symp. on Computer-Based Medical Systems (CBMS*, pages 23–24, 2005.

[18] Yi Liu and Yuan F. Zheng. Minimum enclosing and maximum excluding machine for pattern description and discrimination. *Pattern Recognition, International Conference on*, 3:129–132, 2006.

[19] I. Mierswa, M. Wurst, R. Klinkenberg, M. Scholz, and T. Euler. Yale (now: Rapidminer): Rapid prototyping for complex data mining tasks. In *Proc. of the ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD 2006)*, 2006.

[20] S. Mukkamala, G. Janoski, and A. Sung. Intrusion detection using neural networks and support vector machines. *Proc. of the 2002 Int. Joint Conf. on Neural Networks. IJCNN'02 (Cat. No.02CH37290)*, pages 1702–1707, 2002.

[21] Spiros Papadimitriou, Hiroyuki Kitagawa, Phillip B. Gibbons, and Christos Faloutsos. Loci: Fast outlier detection using the local correlation integral. *Data Engineering, Int. Conf. on*, 0:315, 2003.

[22] Leonid Portnoy, Eleazar Eskin, and Sal Stolfo. Intrusion detection with unlabeled data using clustering. In *In Proc. of ACM CSS Workshop on Data Mining Applied to Security (DMSA-2001)*, pages 5–8, 2001.

[23] Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. In *Proc. of the 2000 ACM SIGMOD Int. Conf. on Management of Data*, SIGMOD '00, pages 427–438, New York, NY, USA, 2000. ACM.

[24] 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*, 13(7):1443–71, July 2001.

[25] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, and John C. Platt. Support vector method for novelty detection. In *Advances in Neural Information Processing Systems 12, (NIPS) Conf.*, pages 582–588. The MIT Press, 11 1999.

[26] Qing Song, Wenjie Hu, and Wenfang Xie. Robust support vector machine with bullet hole image classification. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 32(4):440–448, 2002.

[27] Jian Tang, Zhixiang Chen, Ada Fu, and David Cheung. Enhancing effectiveness of outlier detections for low density patterns. In Ming-Syan Chen, Philip Yu, and Bing Liu, editors, *Advances in Knowledge Discovery and Data Mining*, volume 2336 of *Lecture Notes in Computer Science*, pages 535–548. Springer, 2002.

[28] David M. J. Tax and Robert P. W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[29] Joost van Beusekom and Faisal Shafait. Distortion measurement for automatic document verification. In *Proc. of the 11th Int. Conf. on Document Analysis and Recognition*. IEEE, 9 2011.

[30] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. *Automation and Remote Control*, 24:774–780, 1963.

[31] Vladimir N. Vapnik. *Statistical learning theory*. Wiley, 1 edition, September 1998.

[32] Wenjian Wang, Zongben Xu, Weizhen Lu, and Xiaoyun Zhang. Determination of the spread parameter in the gaussian kernel for classification and regression. *Neurocomputing*, 55(3-4):643–663, October 2003.

[33] Linli Xu, K Crammer, and Dale Schuurmans. Robust support vector machine training via convex outlier ablation. *Proc. of the National Conf. On Artificial Intelligence*, pages 536–542, 2006.

[34] Alan Yuille and Anand Rangarajan. The concave-convex procedure (cccp. In *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[35] Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *J. Mach. Learn. Res.*, 11:1081–1107, March 2010.

[36] Xi-chuan Zhou, Hai-bin Shen, and Jie-ping Ye. Integrating outlier filtering in large margin training. *Journal of Zhejiang University-Science C*, 12(5):362–370, May 2011.