Unsupervised Parameter Estimation for One-Class Support Vector Machines

Zahra Ghafoori[1], Sutharshan Rajasegarar[2], Sarah M. Erfani[1], Shanika Karunasekera[1], and Christopher A. Leckie[1]

[1] Department of Computing and Information Systems,
The University of Melbourne, Australia
[2] School of Information Technology, Deakin University, Australia
{ghafooriz,sarah.erfani,karus,caleckie}@unimelb.edu.edu
sutharshan.rajasegarar@deakin.edu.au

Abstract. Although the hyper-plane based One-Class Support Vector Machine (OCSVM) and the hyper-spherical based Support Vector Data Description (SVDD) algorithms have been shown to be very effective in detecting outliers, their performance on noisy and unlabeled training data has not been widely studied. Moreover, only a few heuristic approaches have been proposed to set the different parameters of these methods in an unsupervised manner. In this paper, we propose two unsupervised methods for estimating the optimal parameter settings to train OCSVM and SVDD models, based on analysing the structure of the data. We show that our heuristic is substantially faster than existing parameter estimation approaches while its accuracy is comparable with supervised parameter learning methods, such as grid-search with cross-validation on labeled data. In addition, our proposed approaches can be used to prepare a labeled data set for a OCSVM or a SVDD from unlabeled data.

Keywords: One-Class Support Vector Machine, Support Vector Data Description, Outlier detection, Parameter estimation

1 Introduction

Abnormal patterns in a data set, which are inconsistent with the majority of the data, are commonly referred to as outliers or anomalies. In many applications, such as fraud detection, environmental monitoring, and medical diagnosis, one of the main tasks is to detect such instances or to remove them [10]. The two major underlying assumptions of many existing outlier detection methods are the rarity of outliers and the distinctive differences between them and the normal data [1].

In general, outlier detection algorithms can be categorized as supervised, semi-supervised or unsupervised learning methods [1, 7]. The former case assumes that both negative and positive labels are available to train a binary classifier, while the latter one does not make any assumption regarding the

availability of a labeled data set [7]. In comparison with these two approaches, semi-supervised methods assume that only the normal examples are available during training, which makes it possible to build a model of normality that rejects anomalous instances [1, 7]. For unsupervised and semi-supervised methods, if it is assumed that the majority of the training data is normal, the methods are also categorised as one-class classification. In this paper, we mainly focus on one-class classification, and interested readers are referred to [1, 7] for more comprehensive surveys.

The OCSVM [14] and SVDD [19] algorithms are two widely used one-class classification methods for outlier detection [11, 4, 15, 6, 9]. It has been shown that the OCSVM and SVDD algorithms handle small fractions of outliers in the training set [19, 14], but if a considerable proportion of such examples exist, both algorithms may end up producing models that are skewed towards outliers [10]. Unfortunately, the availability of a (nearly) clean training data to avoid this problem is not guaranteed in many real applications. Moreover, contributing "good" examples of outliers, i.e., ones that do not lie on normal regions and are far from the normal data points, is sometimes necessary to boost the performance of the OCSVM and SVDD algorithms [19], but we may have no prior knowledge about such examples. Finally, both algorithms have some data dependent parameters whose value can substantially affect the accuracy of the method, and estimating these parameters in an efficient and unsupervised way is an open research problem. Usually, the feature space is searched via grid-search and cross-validation, which are computationally expensive and require labeled examples from both the normal and outlier classes.

This paper addresses the aforementioned problems in the following ways: (i) we propose two fully unsupervised methods to analyse the structure of the data and make a near-optimal estimation of the parameter settings in an efficient way in comparison with existing methods, (ii) we show how our methods can be used to restrict the domain of search in grid-search and improve its efficiency, (iii) we show the application of our proposed methods in pre-processing an unclean data set, comprising a considerable fraction of outliers, and building a labeled data set comprising the normal data and good examples of outliers.

2 Background and Related Work

In this section, a brief explanation of the OCSVM and SVDD algorithms is presented, followed by a review of the related works that have been proposed to find optimal parameter settings for OCSVM or SVDD training.

2.1 One-class Support Vector Machines

The OCSVM or ν-SVM [14] algorithm is a semi-parametric one-class classification method that finds a boundary around dense areas comprising the normal data [7]. In OCSVM, a training set of $x_i \in R^d (i = 1, 2, ..., l)$ feature vectors are projected to a potentially higher dimensional space using a feature map φ. Then,

the algorithm finds a hyper-plane that separates the projected examples from the origin with the maximum possible margin. The primal quadratic problem that the OCSVM classifier solves is as follows:

$$\min_{\omega,\xi,\rho} \frac{1}{2}\|\omega\|^2 + \frac{1}{\nu l}\sum_{i=1}^{l}\xi_i, \;\; s.t.\; (\omega.\varphi(x_i)) \geq \rho - \xi_i; \;\; \xi_i \geq 0 \;\; \forall i, \tag{1}$$

where $\omega \in R^d$ and $0 < \nu \leq 1$. In addition, $\xi_i \geq 0$ are slack variables that relax the problem constraints and allow some examples to fall outside the model boundary. Any given solution for this optimization problem has three separate sets of examples: examples that fall inside the boundary (non-support vectors), examples that lie on the boundary (border support vectors), and examples that fall outside the boundary (outliers or bounded support vectors). One of the important properties of OCSVM is that the user-defined parameter ν is an upper bound on the fraction of outliers and a lower bound on the fraction of support vectors. Using a kernel function as φ, like a Gaussian kernel ($k(x,y) = e^{-\gamma\|x-y\|^2}$) with the kernel parameter γ, it is possible to apply the kernel trick and separate normal data points and outliers that are not linearly separable in the input space. After the training phase, the label of any unseen data x is simply predicted using the decision function $f(x) = sign((\omega.\varphi(x)) - \rho)$.

SVDD [19] has a similar optimization function (Equation 2), but instead of a hyper-plane, it minimizes the radius R of a hyper-sphere that encompasses almost all normal samples:

$$\min_{R,a} R^2 + C\sum_{i=1}^{l}\xi_i, \;\; s.t.\; \|x_i - a\|^2 \leq R^2 + \xi_i; \;\; \xi_i \geq 0 \;\; \forall i, \tag{2}$$

where a is the center of the hyper-sphere and C is a user-defined regularization parameter that has a similar effect as the ν parameter in a OCSVM.

Assuming two training examples x and y, if an applied kernel only depends on $x - y$, i.e., the kernel is stationary, the SVDD and OCSVM algorithms result in equal solutions [14]. As we use the RBF kernel throughout this paper, which is stationary based on the proposed definition, and the ν and C parameters can be defined based on each other using the $\nu = \frac{1}{Cl}$ formula [18], hereafter, we assume that the discussions made for a OCSVM are also valid for a SVDD.

2.2 Estimating Parameter Settings for the OCSVM Algorithm

The choice of the values for γ (the kernel width parameter) and ν (the regularisation parameter) has a major influence on the accuracy of a model generated by the OCSVM or SVDD algorithms [18, 12]. To illustrate the effect of the parameters, we have designed an experiment with a toy problem named Half Kernel, that includes 4,000 normal samples and 5% outliers, which were added to the normal data at random using a uniform distribution. Figure 1 shows the different models that have been generated using the OCSVM algorithm with different values of the γ and ν parameters. The best model, i.e., Figure 1(b), has

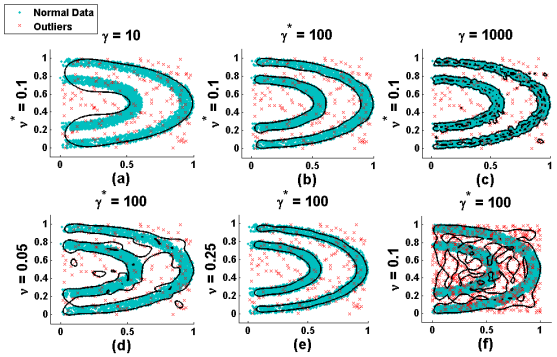

Fig. 1. Sensitivity of a OCSVM to the γ and ν parameters (Half Kernel data set).

been built using the optimal parameter settings (ν^*, γ^*). Figures 1(a) and 1(c) show that choosing γ values less than the optimal value results in building overly general and simple models with a high false positive rate (FPR) with respect to the target class, whereas larger values are prone to building poor models with a high false negative rate (FNR). The ν parameter affects the results in the same manner as shown in Figures 1(d) and 1(e). To illustrate how the optimal value of this parameter depends on the data, in Figure 1(f) we have added another 5% anomalies to the Half Kernel data set and used the same setting as in Figure 1(b) to train a OCSVM model. This figure shows how dramatically the model may deviate towards outliers, if the ν parameter is not set correctly. Consequently, the optimal ν and γ parameter values depend on the given data set, and thus we require a method to select these parameter values from the data. We now summarize two families of approaches to this problem, namely, supervised and unsupervised learning approaches.

Supervised Learning of the Parameters: If ground truth labels are available, it is possible to optimize the choice of values for the ν and γ parameters via n-fold cross-validation. Zhuang et al. [21] suggested that the most reliable approach to search the parameter space in this manner is to use grid-search, and due to its considerable computational requirements, an efficient parameter search algorithm should be used. To this end, they have first applied a coarse-grained search over the entire parameter space and then performed two further fine-grained searches to reduce the complexity by restricting the search space.

Unsupervised Learning of the Parameters: A major drawback of supervised learning in this context is the need for ground truth labels. Consequently, several heuristic unsupervised approaches have been proposed to estimate the γ and ν parameters when the training set is not clean and no ground truth labels are available to find an optimal parameter setting. Emmott et al. [5] have assumed prior knowledge about the value of the ν parameter. Then, the value of the γ parameter has been increased until a predefined proportion of the data has been rejected. Since there is usually more than one pair of parameters that

result in approximately this predefined proportion of outliers, this approach may not be successful in finding an optimal parameter setting. Since the proportion of outliers might be unknown, Rätsch et al. [12] proposed a heuristic to find an appropriate ν value for a OCSVM. Their main assumptions are that outliers are far enough from normal samples and the γ parameter is known, and the idea is to increase ν over the range (0, 1) to find a value that maximizes the separation distance between the normal class and the rejected samples. The distance is defined by the following equation:

$$D_\nu = \frac{1}{N^+} \sum_{f(x)\geq\rho} f(x) - \frac{1}{N^-} \sum_{f(x)<\rho} f(x), \tag{3}$$

where N^+ and N^- are the number of samples in the target and outlier classes, respectively. Rätsch et al. have reported that if there is no clear separation between negative and positive samples, the proposed heuristic may come up with extreme solutions, i.e., 0 or 1. Moreover, the choice of the γ parameter and its effect on finding a good value for the ν parameter has not been discussed in their work. Liu et al. [10] have estimated the γ parameter as $\frac{1}{\gamma} = 2\sum_{i,j=1}^{l} \|x_i - x_j\|^2/l^2$, which can be used in combination with Rätsch's method to estimate both the γ and ν parameters in a fully unsupervised manner. Hereafter, we call this method Duplex Max-margin Model Selection (DMMS) as it is based on the max-margin principle and maximizes the separation between the two classes.

Tax et al. [18] have proposed a heuristic to estimate the γ and ν parameters for a SVDD in a fully unsupervised manner. Their proposed heuristic optimizes the estimated FP and FN rates by solving the following minimization problem:

$$\Lambda(\gamma, \nu) = \frac{|SV|}{N} + \lambda|SV_b|\left(\frac{1}{\gamma^{0.5} s_{max}}\right)^d, \tag{4}$$

where SV and s_{max} represent the set of support vectors and the maximum distance in the training set, respectively. SV_b indicates the set of border support vectors (i.e., those with $0 < \alpha_i < C$), and λ is a regularizer. The first term ($\frac{|SV|}{N}$) is an estimate of the error on the target class, and the second term controls the error on the outlier class. Since the RBF kernel has been used for this heuristic, it is also possible to use the same parameter setting to train a OCSVM. Hereafter, we refer to this heuristic as Duplex Error-minimisation Model Selection (DEMS) as its objective is to minimize FPR and TPR.

There is another work by Liu et al. [10] that uses an unsupervised self-guided soft labeling mechanism to train a one-class classifier, different from the OCSVM and SVDD methods, by applying the soft labels directly in the optimization problem of the studied one-class classification algorithm, which is very different from the aim of this paper and so is not discussed here.

In addition to estimating the parameters, Suvorov et al. [17] and Tax et al. [19] have proposed to use samples from the outlier class directly in the optimization function of a OCSVM or a SVDD to boost the accuracy. However, Tax et al. [19] have shown that choosing "poor" outlier examples, i.e., outliers that

fall inside or very close to the target class, reduces the accuracy of the trained model to be similar to a random classifier. They have also discussed that if only examples from the target class are available, generating synthetic outliers in low density regions can help tighten the data description and enhance accuracy, but an automatic method to generate such examples has not been proposed.

We summarize the shortcomings of the existing methods as follows:

1. Even a moderately high resolution grid-search may incur a substantial number of iterations and high time-complexity. Moreover, the granularity of the search can have a major effect on the final result.
2. In many applications, examples from the outlier class are not available for use in finding an optimal parameter setting via cross-validation. Moreover, it is not assured that a nearly clean data set of normal samples is available during training. These problems have not been studied by the existing approaches.
3. Even if negative examples are available as well as positive ones, based on the work presented by Tax et al. [19], it is not guaranteed that their contribution improves the accuracy of the trained model, unless they are far enough from the target class. None of the existing methods resolves this problem.
4. All the existing unsupervised parameter estimation methods (except DEMS) assume prior knowledge of either γ or ν, but both parameters may be unknown in many applications. This makes it impossible to optimise one parameter based on knowing the other one.
5. The DEMS method is a fully unsupervised method, but it requires a mechanism like grid-search to search over the parameter space and suffers from the time-complexity problem in point 1 above. Moreover, this approach has been examined on only a limited number of data sets.

3 Problem Statement

We are given an unlabeled data set DS comprising unlabeled data points $x_i \in R^d (i = 1, 2, ..., l)$ from the normal and outlier classes. Like [16], we assume that outliers are uniformly distributed in the feature apace. Our aim is to find a compact region in the search space for the parameters γ and ν that contains the optimal parameter settings γ^* and ν^* for a OCSVM with RBF kernel. Once such a compact region has been found, we can either directly estimate the optimal settings, or efficiently apply a grid-search method within this compact region. In this way, we can estimate the optimal parameter settings for training a OCSVM (or SVDD) without requiring ground truth labels or resorting to exhaustive grid-search, even when the fraction of outliers in DS is high.

In addition, we aim to develop a method of pre-processing the data set DS that can: (1) filter "border-line" data points that can affect the accuracy of the learned OCSVM model; and (2) add synthetic outliers in low density regions of nearly clean data sets to enhance the accuracy of the trained OCSVM by generating labeled data sets.

In the following section we propose two unsupervised methods to automatically estimate optimal parameter values γ^* and ν^*, which address the shortcomings that were identified in Section 2 for the existing methods to this problem.

4 Methodology

We divide the problem of finding optimal parameter settings into two steps: (1) estimating the γ parameter, and (2) estimating the ν parameter.

Estimation of γ: Recall from Section 2 that the γ parameter is the bandwidth parameter of the RBF kernel, which acts as a scaling factor to smooth the learned density estimate to reflect the true data density. Lihi et al. [20] proposed a method that estimates a local scaling factor for each sample x_i in an affinity matrix $A \in R^{l \times l}$ ($A_{ij} = exp(-\gamma_i \gamma_j d^2(x_i, x_j))$, where $d(.,.)$ is some distance metric). They used the distance between x_i and its Kth nearest neighbor to obtain an estimate of γ_i, and showed that setting $K = 7$ results in good estimates even for high-dimensional data sets.

Inspired by Lihi et al. [20], we introduce a density measure s_K^i for each data point x_i in the training set DS:

$$s_K^i = \frac{1}{K} \sum_{k \in KNN_i} \|x_i - x_k\|; \;\; \forall i = 1..l, \tag{5}$$

where KNN_i is the set of the K nearest neighbors of x_i inclusive. The density measure s_K^i reflects the density of points around point x_i. Our challenge is to find the value $s_K^{i^*}$ of a point x_{i^*} that corresponds to the "limit" of the density of normal points, and can thus be used to estimate the γ parameter. To do this, we define an ordered set $S_K = \{s_K^i, \forall i = 1..l | S_k^m \geq S_K^{m-1}, \forall m = 1..l\}$, which can be used to fit a function $FS(m)$ to visualize the densities of points in DS. It can be shown that for data sets that follow the definition of DS in Section 3, the function $FS(m)$ is similar to Figure 2. We propose that the knee-point in this monotonically increasing function, which is shown in Figure 2 by a circle, carries important information that can be used to set the γ parameter. The knee-point actually represents a sudden change in the densities where we have the normal points near to the data boundaries followed by the outliers.

For a given monotonically increasing function $f(x)$, a knee-point is a point with maximum curvature. The curvature at each point x of the function $f(x)$ is defined below as $C_f(x)$ [13], hence a knee-point can be formulated as Equation 6.

$$x_{C_f}^{max} = \arg\max_x C_f(x), \text{ where } C_f(x) = \frac{f''(x)}{(1 + f'(x)^2)^{0.5}}. \tag{6}$$

Using this definition, the knee-point of the function $FS(m)$ is $m_{C_{FS}}^{max}$. Thus we set $\gamma = \frac{1}{FS(m_{C_{FS}}^{max})}$ in our heuristic. Later in Section 5, we show that this heuristic works reasonably well for a variety of data sets with different inherent structures. We now propose two variants of our heuristic that provide a means of estimating the ν parameter.

Quick Model Selection (QMS): The information gained via the knee-point of $FS(m)$ can be utilized to estimate the ν parameter as well. As the knee-point is an indicator of a sudden change in $FS(m)$, it shows that densities s_K^i greater than $m_{C_{FS}}^{max}$ are very rare in the training set. As a result, we argue

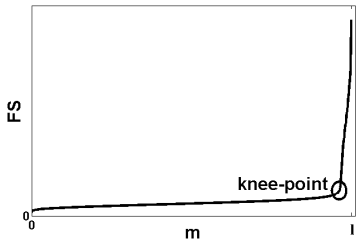

Fig. 2. Illustration of the knee-point of a monotonically increasing function $FS(m)$.

that samples x_i with this property are good representatives of outliers, which leads us to set $\nu = \frac{|S_K| - m^{max}_{C_{FS}}}{|S_K|}$. This is similar to using the K Nearest Neighbor (K-NN) method to detect outliers in a data set, with the key difference that we use this unsupervised method just once in the pre-processing step to estimate an optimal parameter setting for a OCSVM, and after training the OCSVM, unlike the K-NN method, there is no need to compute distances for the test instances.

Another important property of QMS is that it can be used to automatically select good examples of outliers for training purposes. To this end, we introduce a shrinking factor η in the range (0, 1] that can be used to safely divide samples into three groups:

- Normal ($s^i_K < \eta \times FS(m^{max}_{C_{FS}})$)
- Outlier ($s^i_K > (2 - \eta) \times FS(m^{max}_{C_{FS}})$)
- Border-line ($\eta \times FS(m^{max}_{C_{FS}}) \leq s^i_K \leq (2 - \eta) \times FS(m^{max}_{C_{FS}})$)

Now, we can remove the border-line samples from the data set and label the outlier examples as they are sufficiently far from the normal samples. This process is shown in Figure 3 for a Banana data set comprising 10,000 normal instances and 20% anomalies, which were generated using a uniform distribution. This example also illustrates the robustness of our method to the percentage of outliers in the training set.

Revised DMMS (RDMMS): We also propose to use our heuristic to estimate the γ parameter for the DMMS method, which was explained in Section 2. We further modify the distance metric D_ν (Equation 3) as Equation 7 below, for we have found it to be a more practical metric in our experiments:

$$D_\nu = median_{f(x) \geq \rho} f(x) - median_{f(x) < \rho} f(x). \tag{7}$$

Our proposed heuristic approaches can address the challenge of the time-complexity of grid-search, by providing an initial guess for the optimal parameter setting and substantially reducing the search space. In the next section, we evaluate our heuristics on a variety of data sets in terms of their accuracy and run-time, and compare them with supervised grid-search and several existing unsupervised approaches.

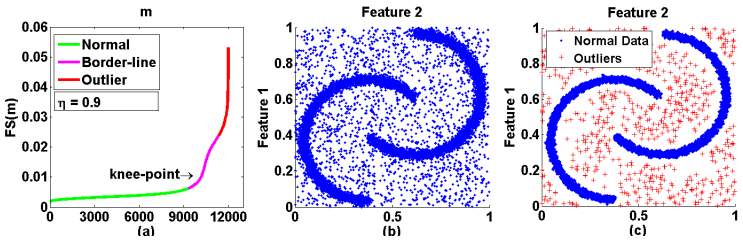

Fig. 3. Pre-processing a Banana data set using QMS; (a) dividing samples into Normal, Border-line, and Outlier sets, (b) the original data set, (c) pruned labeled data set including good examples of outliers.

5 Experimental Evaluation

We evaluated our proposed methods in comparison to the DEMS and DMMS methods. Similar to [20], we set $K = 7$ in our proposed approaches. Since Tax et al. [18] have reported that the DEMS method is not sensitive to the value of the λ parameter, its default value ($\lambda = 1$) was used in our experiments. We also implemented a supervised grid-search method, including two phases of coarse-grained and fine-grained search based on the proposed method by Hsu et al. [8], to make sure that the time-complexity is kept low. This method applies a 10-fold cross validation to find optimal parameter settings.

Several real and synthetic data sets were used to evaluate the accuracy and time-complexity of the methods. To evaluate accuracy, we used the Receiver Operating Characteristic (ROC) curve and the corresponding Area Under the Curve (AUC) as it is insensitive to class balance. The reported AUC values were averaged over 200 runs. The experiments were conducted on a machine with an Intel Core i7CPU at 3.40 GHz and 16 GB RAM. MATLAB LIBSVM toolbox (version 3.20) [2] was used to implement the OCSVM method.

5.1 Data sets

We ran our experiments on 7 real data sets from the UCI Machine Learning Repository, namely Contraceptive Method Choice (CMC), Cardiotocography (Cardio), Breast Cancer Wisconsin (Cancer), Ozone Level Detection (Ozone), Forest CoverType (Forest), Shuttle, and Vowel. We also generated 3 synthetic data sets: a Banana, a C-shape and a Smile (including a mixture of two Gaussians and one C-shaped distribution). These combinations enable us to examine our proposed heuristics on a variety of data structures. All data sets were scaled in the range [0, 1] using feature scaling technique. For all data sets, 5% anomalies in the range [0, 1] were added using a uniform distribution, and test and training sets were randomly selected from the data with the ratio of 1 to 4. In this way we know the actual labels and we are able to evaluate the methods.

We empirically observed that the data should be scaled in the range [0, 1] to make it possible to estimate the γ parameter based on our heuristics.

Table 1. Accuracy and time-complexity of our proposed unsupervised parameter estimation methods (QMS and RDMMS) in comparison with existing supervised (S-Grid-S) and unsupervised (DEMS and DMMS) methods.

		AUC					CPU_Time (in seconds)				
Data set	#Features	DEMS	DMMS	S-Grid-S	**QMS**	**RDMMS**	DEMS	DMMS	S-Grid-S	**QMS**	**RDMMS**
Cancer	10	0.620	0.691	0.796	**0.813**	0.781	2.50	0.42	34.06	**0.04**	0.64
Cardio	22	0.632	0.876	**0.964**	0.958	0.959	31.46	6.65	319.71	**0.27**	7.13
CMC	9	0.522	0.567	0.766	**0.836**	0.805	9.57	2.08	112.40	**0.13**	2.45
Forest	54	0.591	0.883	**0.985**	0.958	0.958	351.93	155.33	10176.60	**7.10**	193.41
Ozone	72	0.697	0.877	**0.981**	0.942	0.941	83.74	14.69	689.87	**0.53**	21.81
Shuttle	9	0.683	0.596	**0.999**	0.995	0.997	236.24	71.17	1555.03	**4.30**	87.93
Vowel	10	0.784	0.867	0.927	**0.957**	**0.951**	3.77	0.70	49.45	**0.06**	0.92
Banana	2	0.850	0.570	**0.900**	0.896	0.849	200.04	59.05	2034.83	**4.21**	145.19
C-shape	2	0.894	0.554	**0.901**	0.898	0.840	196.41	58.34	2277.96	**4.15**	122.85
Smile	20	0.696	0.596	0.981	**0.991**	**0.992**	387.90	122.02	7966.55	**5.29**	148.92
Average		0.697	0.708	0.920	0.924	0.907	150.36	49.05	2521.65	**2.61**	73.13

5.2 Results and Discussion

Table 1 reports the accuracy and run-time of the examined methods. As the traditional supervised grid-search (S-Grid-S) method has considerable computational requirements, we set an upper-bound equal to 10,000 data points on the size of the whole data set in all the experiments reported in this table.

Based on the reported results in Table 1, our proposed methods outperform the existing unsupervised parameter estimation methods (i.e., DEMS and DMMS). In comparison with the S-Grid-S method, on 3 real data sets (Cancer, CMC, and Vowel) our methods result in considerably higher accuracy and lower time complexity, while the S-Grid-S method outperforms our QMS method only on Forest and Ozone, and for the rest of the data sets their accuracy is almost the same. To identify the statistical significance of results between the two approaches with highest AUC, i.e., QMS and S-Grid-S, we conducted a $t-$test with a level of significance of $\alpha = 0.05$. The returned $p = 0.63$ for the accuracy measure fails to reject the null hypothesis with a level of significance, i.e., the difference between the AUC of the two approaches is not statistically significant. Moreover, the returned $p = 0.034$ for the training time indicates that the time-complexity of our method is significantly lower than the S-Grid-S method. Since $l \gg d$ in our experiments, the time-complexity of S-Grid-S using traditional matrix inverse is $O(l^3)$ [3], while the most expensive part of our QMS method, i.e., finding the K nearest neighbors, requires $O(l^2)$. Note that S-Grid-S requires a labeled data set to find the optimal parameter settings, but our proposed methods are completely unsupervised, i.e., learning of the parameter settings is performed without having access to the labels (the labels have been used only for testing purposes).

To compare the scalability of the two methods with the best accuracy, i.e., S-Grid-S and QMS, we have conducted another experiment with the Forest data set. The number of data points has been increased between $10K$ and $500K$, and we have given both methods at most 6 hours to find the optimal parameter settings. As shown in Figure 4, our QMS method successfully finds the parameter

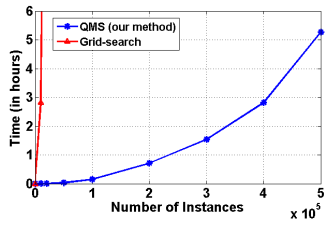

Fig. 4. Scalability of QMS in comparison with the supervised grid-search method on the Forest data set (it took more than 24 hours for the S-Grid-S method to find optimal parameter settings when the size of the data set is $20K$).

settings in this time limit, but the running time of the S-Grid-S method exceeds the limit even for a data set of $20K$ samples.

6 Conclusion

We proposed two parameter estimation algorithms, namely QMS and RDMMS, for the OCSVM and SVDD algorithms, which estimate optimal parameter settings without any need for ground truth labels or exhaustive grid-search over the parameter space. Our experimental evaluation showed that our methods outperformed existing heuristic approaches that found the parameter settings in an unsupervised manner. Moreover, our QMS method had comparable accuracy to the supervised grid-search method, while it was in average more than 900 times faster than the supervised-grid search method on the examined real and synthetic data sets. The QMS method also outperformed all the existing methods in terms of time-complexity. In future work, we aim to use this heuristic in training OCSVMs for concept-drifting data streams, where we need to train a new model using the recent data.

References

[1] Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Computing Surveys (CSUR) 41(3), 15 (2009)
[2] Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST) 2(3), 1–27 (2011)
[3] Chapelle, O.: Training a support vector machine in the primal. Neural Computation 19(5), 1155–1178 (2007)
[4] Chen, Y., Zhou, X.S., Huang, T.S.: One-class SVM for learning in image retrieval. In: Proceedings of the International Conference on Image Processing. vol. 1, pp. 34–37. IEEE (October 2001)
[5] Emmott, A.F., Das, S., Dietterich, T., Fern, A., Wong, W.K.: Systematic construction of anomaly detection benchmarks from real data. In: Proceedings of the ACM SIGKDD Workshop on Outlier Detection and Description. pp. 16–21. ACM (2013)

[6] Heller, K., Svore, K., Keromytis, A.D., Stolfo, S.: One class support vector machines for detecting anomalous windows registry accesses. In: Proceedings of the ICDM Workshop on Data Mining for Computer Security (DMSEC). pp. 2–9. Melbourne, FL, USA (November 2003)
[7] Hodge, V.J., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review 22(2), 85–126 (2004)
[8] Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification (2003)
[9] Hu, W., Liao, Y., Vemuri, V.R.: Robust support vector machines for anomaly detection in computer security. In: Proceedings of the 2003 International Conference on Machine Learning and Applications (ICMLA). pp. 168–174. LA, CA, USA (June 2003)
[10] Liu, W., Hua, G., Smith, J.R.: Unsupervised one-class learning for automatic outlier removal. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3826–3833. IEEE (2014)
[11] Mukkamala, S., Janoski, G., Sung, A.: Intrusion detection using neural networks and support vector machines. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN). vol. 2, pp. 1702–1707. IEEE (May 2002)
[12] Rätsch, G., Mika, S., Scholkopf, B., Müller, K.R.: Constructing boosting algorithms from svms: an application to one-class classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(9), 1184–1199 (2002)
[13] Satopää, V., Albrecht, J., Irwin, D., Raghavan, B.: Finding a "kneedle" in a haystack: Detecting knee points in system behavior. In: Proceedings of the 31st International Conference on Distributed Computing Systems Workshops (ICDCSW). pp. 166–171. IEEE (2011)
[14] Schölkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation 13(7), 1443–1471 (2001)
[15] Shin, H.J., Eom, D.H., Kim, S.S.: One-class support vector machines: an application in machine fault detection and classification. Computers & Industrial Engineering 48(2), 395–408 (2005)
[16] Subramaniam, S., Palpanas, T., Papadopoulos, D., Kalogeraki, V., Gunopulos, D.: Online outlier detection in sensor data using non-parametric models. In: Proceedings of the 32nd international conference on Very large data bases. pp. 187–198. VLDB Endowment (2006)
[17] Suvorov, M., Ivliev, S., Markarian, G., Kolev, D., Zvikhachevskiy, D., Angelov, P.: Osa: One-class recursive svm algorithm with negative samples for fault detection. In: Proceedings of the 23rd International Conference on Artificial Neural Networks (ICANN), pp. 194–207. Springer (2013)
[18] Tax, D.M., Duin, R.P.: Outliers and data descriptions. In: Proceedings of the 7th Annual Conference of the Advanced School for Computing and Imaging. pp. 234–241 (2001)
[19] Tax, D.M., Duin, R.P.: Support vector data description. Machine learning 54(1), 45–66 (2004)
[20] Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Proceedings of Advances in Neural Information Processing Systems (NIPS). pp. 1601–1608 (2004)
[21] Zhuang, L., Dai, H.: Parameter estimation of one-class svm on imbalance text classification. In: Advances in Artificial Intelligence, pp. 538–549. Lecture Notes in Computer Science, Springer (2006)