Simple Incremental One-Class Support Vector Classification

Kai Labusch, Fabian Timm, and Thomas Martinetz

Institute for Neuro- and Bioinformatics, University of Lübeck,
Ratzeburger Allee 160, D-23538 Lübeck, Germany

Abstract. We introduce the OneClassMaxMinOver (OMMO) algorithm for the problem of one-class support vector classification. The algorithm is extremely simple and therefore a convenient choice for practitioners. We prove that in the hard-margin case the algorithm converges with $\mathcal{O}(1/\sqrt{t})$ to the maximum margin solution of the support vector approach for one-class classification introduced by Schölkopf et al. Furthermore, we propose a 2-norm soft margin generalisation of the algorithm and apply the algorithm to artificial datasets and to the real world problem of face detection in images. We obtain the same performance as sophisticated SVM software such as libSVM.

1 Introduction

Over the last years, the support vector machine [1] has become a standard approach in solving pattern recognition tasks. There are several training techniques available, e.g. SMO [2]. Although these methods seem to be simple, they are hard to understand without a background in optimisation theory. Hence, they are difficult to motivate when explained to practitioners. In many cases, in particular in industrial contexts, where external libraries or other third party software cannot be used due to various reasons, these techniques are not applied, even though they might be beneficial for solving the problem.

In many applications one has to cope with the problem that only samples of one class are given. The task is to separate this class from the *other* class that consists of all outliers. Either only few samples of the outlier class are given or the outlier class is missing completely. In these cases two-class classifiers often show bad generalisation performance and it is advantageous to employ one-class classification.

Approaches to one-class classification can be divided into three groups: density estimators, reconstruction methods, and boundary methods. The first and the second group are the most powerful because they derive a model of the data that is defined everywhere in the input space. An advantage of the boundary methods is that they consider an easier problem, that is, describing only the class boundaries, instead of describing the whole distribution of the data.

In the present work, we describe a very simple and incremental boundary method based on the support vector approach. It provides the same solution

as comparable techniques such as SMO, despite being extremely simple and therefore applicable for practitioners who are not within the field of machine learning.

2 Previous Work

In the context of one-class classification several boundary methods have been developed. We only want to give a brief description of two approaches that have been introduced almost simultaneously. Tax et al [3] consider the problem of finding the smallest enclosing ball of given data samples $\boldsymbol{x}_i \in \mathcal{X}, i = 1, \ldots, L$ that is described by the radius R and centre $\boldsymbol{w}$:

$$\min_{\boldsymbol{w},R} \left(R + \frac{1}{\nu l} \sum_i \xi_i \right) \quad \text{s.t. } \forall i : \|\phi(\boldsymbol{x}_i) - \boldsymbol{w}\| \leq R - \xi_i \ \wedge \ \xi_i \geq 0 \ . \tag{1}$$

This is the soft version of the problem. It deals with outliers by using slack variables ξ_i in order to allow for samples that are not located inside the ball defined by $\boldsymbol{w}$ and R. For $\nu \to 0$ one obtains the hard-margin solution, where all samples are located inside the ball. Here ϕ denotes a mapping of the data samples to some feature space.

Schölkopf et. al [4] show that one-class classification can be taken as two-class classification, where the *other* class is represented by the origin. They consider the problem of finding the hyperplane $\boldsymbol{w}$ that separates the data samples from the origin with maximum distance ρ:

$$\min_{\boldsymbol{w},\boldsymbol{\xi},\rho} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{1}{\nu l} \sum_i \xi_i - \rho \right) \quad \text{s.t. } \forall i : \boldsymbol{w}^T \phi(\boldsymbol{x}_i) \geq \rho - \xi_i \ \wedge \ \xi_i \geq 0 \ . \tag{2}$$

Again, the soft-margin problem is shown. It allows for samples that are misclassified by using slack variables ξ_i. For $\nu \to 0$ one obtains the hard-margin solution that enforces correct classification of all given samples.

In [4] it is shown that (1) and (2) turn out to be equivalent, if the $\phi(\boldsymbol{x}_i)$ lie on the surface of a sphere. Then, the radius R of problem (1) and the margin ρ in (2) can easily be computed by choosing a support vector on the boundary. If the Gaussian kernel is used

$$K(\boldsymbol{x}, \boldsymbol{y}) = \langle \phi(\boldsymbol{x}), \phi(\boldsymbol{y}) \rangle = \exp\left(-\|\boldsymbol{x} - \boldsymbol{y}\|/2\sigma^2\right) \tag{3}$$

in order to implicitly map the given samples to some feature space, the $\phi(\boldsymbol{x}_i)$ have unit norm and this condition is satisfied. To make the problem solvable, the origin has to be linearly separable from the target class. This precondition is also given if a Gaussian kernel is used. In the following we require that the data has been mapped to some feature space where these conditions hold, i.e. linear separability of the origin and unit norm of all samples.

3 OneClassMaxMinOver

In this section we describe a very simple incremental algorithm for one-class-classification called OneClassMaxMinOver(OMMO). This algorithm is closely connected to problem (2). It is inspired by the MaxMinOver algorithm for two-class classification proposed in [5]. We consider the problem of finding the hyperplane $\boldsymbol{w}_*$ passing the origin and having maximum margin ρ_* with respect to the given data samples. Finding this hyperplane is equivalent to solving the optimisation problem (2), that is finding the hyperplane $\boldsymbol{w}_*$ that separates the given data samples with maximum margin ρ_* from the origin (see Fig. 1).

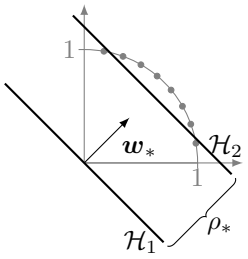


Fig. 1. Some data samples having unit norm are shown as well as the solutions of the optimisation problem (4), i.e. $\mathcal{H}_1$ and the solution of (2), i.e. $\mathcal{H}_2$.

Mathematically, we are looking for the solution of the following optimisation problem

$$\boldsymbol{w}_* = \arg\max_{\boldsymbol{w}} \left(\min_{\boldsymbol{x}_i} \left(\boldsymbol{w}^T \boldsymbol{x}_i \right) \right) \quad \text{s.t. } \|\boldsymbol{w}\| = 1 \ . \tag{4}$$

The margin ρ_* is obtained by

$$\rho_* = \min_{\boldsymbol{x}_i} \left(\boldsymbol{w}_*^T \boldsymbol{x}_i \right) \ . \tag{5}$$

In the following $\boldsymbol{w}_t$ denotes the approximation of $\boldsymbol{w}_*$ at time t. During the learning process the constraint $\|\boldsymbol{w}\| = 1$ is dropped. The algorithm starts with $\boldsymbol{w}_0 = \boldsymbol{0}$ and after $t_{\max}$ learning iterations the norm of the final approximation $\boldsymbol{w}_{t_{\max}}$ is set to one. In each learning iteration, the algorithm selects the sample that is closest to the current hyperplane defined by $\boldsymbol{w}_t$

$$\boldsymbol{x}_{\min}(t) = \arg\min_{\boldsymbol{x}_i} \left(\boldsymbol{w}_t^T \boldsymbol{x}_i \right) \ . \tag{6}$$

For each given training sample $\boldsymbol{x}_i$ there is a counter variable α_i that is increased by 2 whenever the sample is selected as $\boldsymbol{x}_{\min}(t)$:

$$\alpha_i = \alpha_i + 2 \quad \text{for} \quad \boldsymbol{x}_{\min}(t) = \boldsymbol{x}_i \ . \tag{7}$$

$\mathcal{X}'(t)$ denotes the set of samples $\boldsymbol{x}_j$ for which $\alpha_j > 0$ holds at time t. Out of this set, the algorithm selects the sample being most distant with respect to the current hyperplane defined by $\boldsymbol{w}_t$:

$$\boldsymbol{x}_{\max}(t) = \arg\max_{\boldsymbol{x}_j \in \mathcal{X}'(t)} \left(\boldsymbol{w}_t^T \boldsymbol{x}_j \right) \ . \tag{8}$$

Whenever a sample is selected as $\boldsymbol{x}_{\max}(t)$, its associated counter variable is decreased by 1:

$$\alpha_i = \alpha_i - 1 \quad \text{for} \quad \boldsymbol{x}_{\max}(t) = \boldsymbol{x}_i \ . \tag{9}$$

The approximation of $\boldsymbol{w}_*$ in learning iteration $t+1$ is given by

$$\boldsymbol{w}_{t+1} = \sum_{i=1}^{L} \alpha_i \boldsymbol{x}_i \ . \tag{10}$$

Note that (7) and (9) can be combined to the learning rule

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + 2\boldsymbol{x}_{\min}(t) - \boldsymbol{x}_{\max}(t) \ . \tag{11}$$

Altogether, we obtain algorithm 1 for incremental one-class classification.

Algorithm 1: OneClassMaxMinOver. With $h(\boldsymbol{x}_i) = \sum_{j=1}^{L} \alpha_j \boldsymbol{x}_j^T \boldsymbol{x}_i$

$\alpha_i \leftarrow 0 \quad \forall i = 1, ..., N$
for $t = 0$ to $t_{\max}$ **do**
  $\boldsymbol{x}_{\min}(t) \leftarrow \underset{\boldsymbol{x}_i \in \mathcal{X}}{\arg\min}\, h(\boldsymbol{x}_i)$
  $\boldsymbol{x}_{\max}(t) \leftarrow \underset{\boldsymbol{x}_i \in \mathcal{X}'(t)}{\arg\max}\, h(\boldsymbol{x}_i)$
  $\alpha_{\min} \leftarrow \alpha_{\min} + 2$
  $\alpha_{\max} \leftarrow \alpha_{\max} - 1$
end
$\boldsymbol{\alpha} \leftarrow \boldsymbol{\alpha} / \sqrt{\sum_i \alpha_i h(\boldsymbol{x}_i)}$
$\rho \leftarrow \min_{\boldsymbol{x}_i \in \mathcal{X}} h(\boldsymbol{x}_i)$

In section 3.1 we are going to prove that for $t \to \infty$ the following propositions hold:

- $\boldsymbol{w}_t / \|\boldsymbol{w}_t\|$ converges at least as $\mathcal{O}(1/\sqrt{t})$ to $\boldsymbol{w}_*$.
- $\alpha_i > 0$ only holds for samples $\boldsymbol{x}_i$ having distance ρ_* with respect to the hyperplane $\boldsymbol{w}_*$, i.e. support vectors.

As mentioned before, we require that the data set has been mapped into some feature space where all samples have unit norm and can be linearly separated from the origin. However, this has not to be done explicitly. It can also be achieved by replacing the standard scalar product with a kernel that implements an implicit mapping to a feature space having the required properties. In case of the OMMO algorithm this corresponds to replacing the function $h(\boldsymbol{x}_i)$ with

$$h(\boldsymbol{x}_i) = \sum_{j=1}^{L} \alpha_j K(\boldsymbol{x}_j, \boldsymbol{x}_i) \quad , \tag{12}$$

where $K(\boldsymbol{x}_j, \boldsymbol{x}_i)$ is an appropriate kernel function. For a discussion of kernel functions see, for example, [6].

3.1 Proof of Convergence

Our proof of convergence for the OneClassMaxMinOver algorithm is based on the proof of convergence for MaxMinOver by Martinetz [5], who showed a convergence speed of $\mathcal{O}(1/\sqrt{t})$ in the case of two-class classification.

Proposition 1. *The length of $\boldsymbol{w}_t$ is bounded such that $\|\boldsymbol{w}_t\| \leq \rho_* \, t + 3\sqrt{t}$.*

Proof. This is done by induction and using the properties that

$$\rho_* \geq \rho_t = \frac{\boldsymbol{w}_t^T \boldsymbol{x}_{\min}(t)}{\|\boldsymbol{w}_t\|} \Rightarrow \rho_* \|\boldsymbol{w}_t\| \geq \boldsymbol{w}_t^T \boldsymbol{x}_{\min}(t) \ , \tag{13}$$

$$\forall \ \ i: \ \|\boldsymbol{x}_i\| = 1 \ , \text{ and} \tag{14}$$

$$\forall \ \ t: \ \boldsymbol{x}_{\min}(t)^T \boldsymbol{x}_{\max}(t) = \cos\beta \ \underbrace{\|\boldsymbol{x}_{\min}(t)\|}_{=1} \ \underbrace{\|\boldsymbol{x}_{\max}(t)\|}_{=1} \geq -1 \ . \tag{15}$$

The case $t = 0$ is trivial and for $t \to t+1$ it follows that

$$\begin{aligned}
\|\boldsymbol{w}_{t+1}\|^2 &\overset{(11)}{=} \boldsymbol{w}_t^T \boldsymbol{w}_t + 2\boldsymbol{w}_t^T \left(2\boldsymbol{x}_{\min}(t) - \boldsymbol{x}_{\max}(t)\right) + \left(2\boldsymbol{x}_{\min}(t) - \boldsymbol{x}_{\max}(t)\right)^2 \\
&= \boldsymbol{w}_t^T \boldsymbol{w}_t + 2\boldsymbol{w}_t^T \boldsymbol{x}_{\min}(t) + 2 \underbrace{\left(\boldsymbol{w}_t^T \boldsymbol{x}_{\min}(t) - \boldsymbol{w}_t^T \boldsymbol{x}_{\max}(t)\right)}_{\leq 0} + \\
&\quad 4\boldsymbol{x}_{\min}(t)^T \boldsymbol{x}_{\min}(t) + \boldsymbol{x}_{\max}(t)^T \boldsymbol{x}_{\max}(t) - 4\boldsymbol{x}_{\min}(t)^T \boldsymbol{x}_{\max}(t) \\
&\overset{(14),(15)}{\leq} \boldsymbol{w}_t^T \boldsymbol{w}_t + 2\boldsymbol{w}_t^T \boldsymbol{x}_{\min}(t) + 9 \\
&\overset{(13)}{\leq} \boldsymbol{w}_t^T \boldsymbol{w}_t + 2\rho_* \|\boldsymbol{w}_t\| + 9 \\
&\leq \left(\rho_* \, t + 3\sqrt{t}\right)^2 + 2\rho_* \left(\rho_* \, t + 3\sqrt{t}\right) + 9 \\
&= \rho_*^2 \, t^2 + 2\rho_*^2 \, t + 6\rho_* \, t\sqrt{t} + 6\rho_* \sqrt{t} + 9t + 9 \\
&\leq \rho_*^2 \left(t^2 + 2t\right) + 6\rho_* \, (t+1)\sqrt{t} + 9(t+1) + \rho_*^2 \\
&\leq \rho_*^2 \, (t+1)^2 + 6\rho_* \, (t+1)\sqrt{t+1} + 9(t+1) \\
&= \left(\rho_* \, (t+1) + 3\sqrt{t+1}\right)^2 \ .
\end{aligned}$$

□

Theorem 1. *For $t \to \infty$ the angle γ_t between the optimal direction $\boldsymbol{w}_*$ and the direction $\boldsymbol{w}_t$ found by OMMO converges to zero, i.e. $\lim_{t\to\infty} \gamma_t = 0$.*

Proof.

$$\begin{aligned}
\cos\gamma_t &= \frac{\boldsymbol{w}_*^T \boldsymbol{w}_t}{\|\boldsymbol{w}_t\|} \\
&= \frac{1}{\|\boldsymbol{w}_t\|} \sum_{i=0}^{t-1} \boldsymbol{w}_*^T \left(2 \, \boldsymbol{x}_{\min}(i) - \boldsymbol{x}_{\max}(i)\right)
\end{aligned} \tag{16}$$

$$
\begin{aligned}
&= \frac{1}{\|\boldsymbol{w}_t\|} \sum_{i=0}^{t-1} \underbrace{\boldsymbol{w}_*^T \boldsymbol{x}_{\min}(i)}_{\geq \rho_*} \;\geq\; \frac{1}{\|\boldsymbol{w}_t\|} \rho_* \, t \qquad (17)\\
&\overset{\text{Prop.1}}{\geq} \frac{\rho_* \, t}{\rho_* \, t + 3\sqrt{t}} = \frac{1}{1 + \frac{3\sqrt{t}}{\rho_* \, t}}\\
&\geq 1 - \frac{3}{\rho_* \sqrt{t}} \;\overset{t \to \infty}{\longrightarrow}\; 1
\end{aligned}
$$

From (16) to (17) we have used that a sample can only be *forgotten*, if it was *learnt* before, such that $\forall \, \boldsymbol{x}_{\max}(t) \; \exists \, \boldsymbol{x}_{\min}(t') \, , \; t' < t \; : \; \boldsymbol{x}_{\max}(t) = \boldsymbol{x}_{\min}(t')$. □

Theorem 2. *Beyond some finite number of iterations $\hat{t}$ the set $\mathcal{X}'(t)$ will always consist only of support vectors.*

Proof. First, we show that after some finite number of iterations t' the $\boldsymbol{x}_{\min}(t)$ with $t > t'$ will always be a support vector. We use an orthogonal decomposition of $\boldsymbol{w}_t$ as shown in Fig. 2. $\mathcal{X}^{\text{sv}}$ will denote the set of true support vectors.
For an indirect proof we assume that a finite number of iterations t' where $\boldsymbol{x}_{\min}(t)$ with $t > t'$ will always be a support vector does not exist, i.e.
$\nexists \, t' < \infty \;\; \forall \, t \, , \; t' < t : \; \boldsymbol{x}_{\min}(t) \in \mathcal{X}^{\text{sv}}$.

$$
\begin{aligned}
\Rightarrow \rho_* \geq \rho_t \;&=\; \frac{\boldsymbol{w}_t^T \boldsymbol{x}_{\min}(t)}{\|\boldsymbol{w}_t\|} \;\overset{(20)}{=}\; \frac{\left(\cos\gamma_t \|\boldsymbol{w}_t\| \boldsymbol{w}_* + \boldsymbol{u}_t\right)^T \boldsymbol{x}_{\min}(t)}{\|\boldsymbol{w}_t\|}\\
&=\; \cos\gamma_t \boldsymbol{w}_*^T \boldsymbol{x}_{\min}(t) + \frac{\boldsymbol{u}_t^T \boldsymbol{x}_{\min}(t)}{\|\boldsymbol{w}_t\|}\\
&\overset{(21)}{=} \underbrace{\cos\gamma_t}_{\overset{t\to\infty}{\longrightarrow} 1} \underbrace{\boldsymbol{w}_*^T \boldsymbol{x}_{\min}(t)}_{\geq \rho_*} + \underbrace{\frac{\boldsymbol{u}_t^T \boldsymbol{x}_{\min}(t)}{\|\boldsymbol{u}_t\|}}_{\leq 1} \underbrace{\sin\gamma_t}_{\overset{t\to\infty}{\longrightarrow} 0} \qquad (18)
\end{aligned}
$$

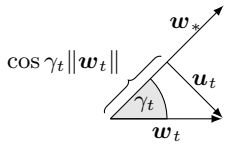
If $\boldsymbol{x}_{\min}(t)$ is not a support vector, $\boldsymbol{w}_* \boldsymbol{x}_{\min}(t) > \rho_*$ holds. Due to (18), there is a t' where $\boldsymbol{x}_{\min}(t)$ being a non-support vector and $t > t'$ inevitably leads to a contradiction. Note that for $t > t'$ only support vectors are added to the set $\mathcal{X}'(t)$, i.e. there is a finite number of non-support vectors contained in the set $\mathcal{X}'(t)$. As a consequence after a finite number of iterations t'' also $\boldsymbol{x}_{\max}(t)$ will always be a support vector.

Now, we show that all non-support vectors in the set $\mathcal{X}'(t)$ will be removed. *Assumption*: There exists a sample $\boldsymbol{x}$ that is not a support vector but it remains in the set $\mathcal{X}'(t)$, i.e. $\exists \, \boldsymbol{x} : \boldsymbol{x} \notin \mathcal{X}^{\text{sv}} \wedge \boldsymbol{x} \in \mathcal{X}'(t)$ for all t. This means that

$$
\frac{\boldsymbol{w}_t}{\|\boldsymbol{w}_t\|} \boldsymbol{x} < \underbrace{\frac{\boldsymbol{w}_t}{\|\boldsymbol{w}_t\|} \boldsymbol{x}_{\max}(t)}_{\overset{t\to\infty}{\longrightarrow} \rho_*} \qquad (19)
$$

always holds. This leads to a contradiction since after a finite number of iterations $\boldsymbol{x}_{\max}(t)$ will always be a support vector.

□



$$\boldsymbol{w}_t = \cos\gamma_t \|\boldsymbol{w}_t\| \boldsymbol{w}_* + \boldsymbol{u}_t \tag{20}$$
$$\|\boldsymbol{u}_t\| = \|\boldsymbol{w}_t\| \sin\gamma_t \tag{21}$$
$$\|\boldsymbol{w}_*\| = 1$$

Fig. 2. Orthogonal decomposition of $\boldsymbol{w}_t$ and properties that hold within this decomposition.

3.2 Soft-OneClassMaxMinOver

So far, we only have considered the hard-margin problem. In order to realise a 2-norm soft-margin version of the OMMO algorithm, we consider the quadratic optimisation problem:

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \left(\frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{2}\sum_i \xi_i^2 \right) \quad \text{s.t.} \quad \forall i : \boldsymbol{w}^T\phi(\boldsymbol{x}_i) \geq 1 - \xi_i \quad . \tag{22}$$

In the hard-margin case ($C \to \infty$) this is equivalent to the optimisation problem (4). Note, that compared to the optimisation problems (1) and (2) the constraint on each slack variable ($\xi_i \geq 0$) disappears. By constructing the primal Lagrangian of (22), setting the partial differentiations to zero and rearranging [6], we obtain

$$\min_{\boldsymbol{\alpha}} \left(\sum_i \alpha_i - \frac{1}{2}\sum_{i,j} \alpha_i\alpha_j \left(K(\boldsymbol{x}_i,\boldsymbol{x}_j) + \frac{1}{C}\delta_{ij} \right) \right) \quad \text{s.t.} \ \ \forall i : \ \alpha_i \geq 0 \ , \tag{23}$$

where δ_{ij} is the Kronecker delta which is 1 if $i = j$, and 0 otherwise. As mentioned in [6] this can be understood as solving the hard-margin problem in a modified kernel space. The modified kernel is $K(\boldsymbol{x}_i,\boldsymbol{x}_j) + \frac{1}{C}\delta_{ij}$. Hence, in order to implement a 2-norm soft-margin version of OMMO, we modify algorithm 1 such that

$$h(\boldsymbol{x}_i) = \sum_{j=1}^{L} \alpha_j \left(K(\boldsymbol{x}_j,\boldsymbol{x}_i) + \frac{1}{C}\delta_{ij} \right) \quad . \tag{24}$$

4 Experiments & Results

We applied the OMMO algorithm to artificial datasets and a real-world problem using Gaussian kernels. We created a sinusoid and an xor dataset each consisting of 250 samples (Fig. 3). The hyperparameters were set to extremal values and to more appropriate ones that can be determined, for instance, by cross-validation. The results on the artificial datasets are shown in Fig. 3. Similar to the approach (2) that implements a 1-norm slack term, different solutions ranging from hard to soft margin can be realised by controlling the parameter C, i.e. the relevance

of outliers can be controlled. Furthermore, we applied the OMMO algorithm to the problem of face detection where we used the MIT-CBCL face detection dataset[1] that contains 2901 images of faces and 28121 images of non-faces of size 19x19 pixels. The dataset is divided into a training set containing 2429 faces and 4548 non-faces and a test set containing 472 faces and 23573 non-faces. We used the raw data but performed the preprocessing steps described in [7] to reduce the within-class variance. Afterwards, we took the training set to perform a simple grid search over σ, C and chose randomly 1215 faces to train OMMO and tested the performance on a test set with 1214 faces and 4548 non-faces. To reduce the variance we performed 25 runs at all combinations of σ, C. The performance of OMMO for a fixed σ, C was evaluated by the equal-error-rate of the receiver-operator-characteristics (ROC). Having determined the optimal parameters σ, C, we trained OMMO with the whole training set of 2429 faces and computed the ROC curve of the 24045 test samples. The same steps were performed using the libSVM [8], except that here we used the parameter ν to control the softness. A comparison between both ROC curves is depicted in Fig. 4. Although these two approaches differ significantly in their implementation complexity, their performance is almost equal. The execution time of OMMO and libSVM cannot be compared directly because it depends heavily on implementation details.

5 Conclusions

Based on an existing two-class classification method, we proposed a very simple and incremental boundary approach, called **O**neClass**M**ax**M**in**O**ver. OMMO can be realised by only a few lines of code,which makes it interesting particularly for practitioners. We proved that after a finite number of learning steps OMMO yields a maximum margin hyperplane that is described only by support vectors. Furthermore, we showed that the speed of convergence of OMMO is $\mathcal{O}(1/\sqrt{t})$ where t is number of iterations. Considering the ideas described in [9], even a $\mathcal{O}(1/t)$ convergence of OMMO can be expected.

By simply using a modified kernel function $K'(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x}, \boldsymbol{y}) + \frac{1}{C}\delta_{xy}$ OMMO can also realise a soft maximum margin solution controlled by the softness parameter C. Thus, OMMO can cope with datasets that also contain outliers.

In the future, a closer look at convergence speed and bounds on the target error will be made. Moreover, the problem of parameter validation will be examined, since in many one-class problems only target objects are available. Thus, standard validation techniques cannot be applied. If it is not possible to evaluate the target error, the complexity or volume of the boundary description have to be estimated in order to select good hyperparameters. Since, simple sampling techniques fail to measure the volume in high-dimensional input spaces, more sophisticated methods need to be derived.

[1] http://cbcl.mit.edu/software-datasets/FaceData.html

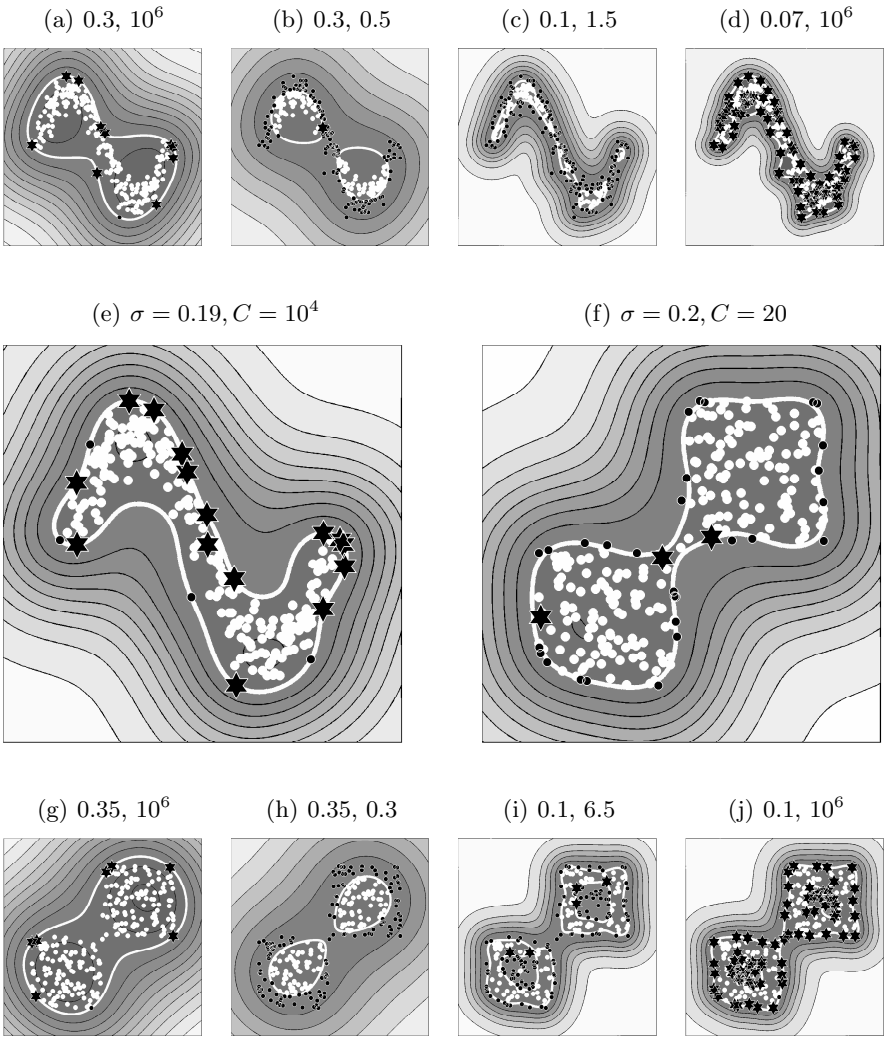


Fig. 3. Our algorithm applied to two exemplary artificial datasets – sinusoid, and xor. The parameters (σ, C) are shown above each graph. The stars depict support vectors that lie inside the hypersphere, dark circles depict support vectors outside the hypersphere. All other samples as well as the boundary are represented in white. In the first and third row extremal values for σ and C were chosen to achieve hard-margin ((a), (d), (g), (j)) as well as soft margin solutions ((b), (c), (h), (i)). In (b), (c), and (h) there is no support vector which lies inside the hypersphere and so the boundary is only influenced by support vectors from outside the hypersphere. The best solutions for the datasets are shown in the second row, where the values lie in between the extremal hard and soft margin solutions.

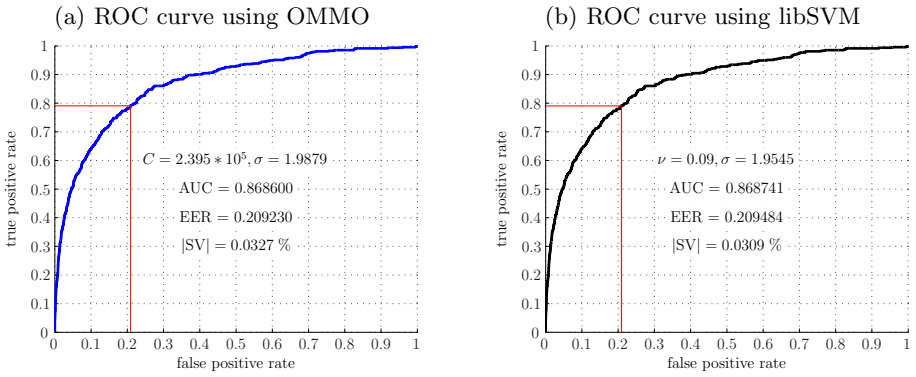


Fig. 4. The receiver-operator-characteristics shows that the two algorithms achieve the same performance on a test set of 472 faces and 23573 non-faces. Both models obtained by parameter validation are rather hard- than soft-margin (C large, ν small). They have a Gaussian width of $\sigma \approx 1.9$ and the fraction of support vectors is almost equal. The performance measured by the area under curve (AUC) for OMMO is the same as for libSVM. This holds also for the equal-error-rate (EER).

References

1. Vapnik, V.N.: The nature of statistical learning theory. Springer Verlag, Heidelberg, DE (1995)
2. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B., Burges, C.J.C., Smola, A.J., eds.: Advances in Kernel Methods — Support Vector Learning, Cambridge, MA, MIT Press (1999) 185–208
3. Tax, D.M.J., Duin, R.P.W.: Data domain description using support vectors. In: ESANN. (1999) 251–256
4. 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) (2001) 1443–1471
5. Martinetz, T.: MaxMinOver: A Simple Incremental Learning Procedure for Support Vector Classification. In: IEEE Proceedings of the International Joint Conference on Neural Networks (IJCNN 2004), Budapest, Hungary (2004) 2065–2070
6. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge, U.K. (2000)
7. Sung, K.K.: Learning and example selection for object and pattern detection. In: MIT AI-TR. (1996)
8. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. (2001) Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.
9. Martinetz, T.: Minover revisited for incremental support-vector-classification. In Rasmussen, C.E., Bülthoff, H.H., Schölkopf, B., Giese, M.A., eds.: DAGM-Symposium. Volume 3175 of Lecture Notes in Computer Science., Springer (2004) 187–194