Multi-Class Supervised Novelty Detection

Vilen Jumutc and Johan A.K. Suykens, *Senior member, IEEE*

***Abstract*—In this paper we study the problem of finding a support of unknown high-dimensional distributions in the presence of labeling information, called Supervised Novelty Detection (SND). The One-Class Support Vector Machine (SVM) is a widely used kernel-based technique to address this problem. However with the latter approach it is difficult to model a mixture of distributions from which the support might be constituted. We address this issue by presenting a new class of SVM-like algorithms which help to approach multi-class classification and novelty detection from a new perspective. We introduce a new coupling term between classes which leverages the problem of finding a good decision boundary while preserving the compactness of a support with the l_2-norm penalty. First we present our optimization objective in the primal and then derive a dual QP formulation of the problem. Next we propose a Least-Squares formulation which results in a linear system which drastically reduces computational costs. Finally we derive a Pegasos-based formulation which can effectively cope with large datasets that cannot be handled by many existing QP solvers. We complete our paper with experiments that validate the usefulness and practical importance of the proposed methods both in classification and novelty detection settings.**

***Index Terms*—Novelty detection, One-Class SVM, classification, pattern recognition, labeling information.**

I. Introduction

Novelty or anomaly detection is a widely recognized machine learning problem where one tries to find a compact support of some unknown probability distribution. Many existing methods, like One-Class SVM [1] or Bayesian approaches [2], heavily rely on the *i.i.d.* assumption and deal with unlabeled data. Contrary to these methods it was proposed recently [3] to approach novelty detection from a classification perspective. In this setting one tries to tackle density estimation via a weighted binary classification problem. However, while the results presented in [3] are consistent with those obtained by other works on Novelty Detection [4], [5], it is still unclear how these methods behave when the *i.i.d.* assumption does not hold or data are generated by a mixture of distributions. In this research we try to close the gap by answering some of the following questions. What if we model the support of each distribution (class) separately? How, in this case, are these models relating to each other? What is the optimal interpretation of such a problem?

In this paper we concentrate on presenting three different extensions of our previous method of Supervised Novelty Detection (SND) introduced in [6]. The first extension is formulated in terms of a QP problem with box constraints. The second one is a Least-Squares problem given by a linear Karush-Kuhn-Tucker (KKT) system. The third one is related to large-scale problems where one cannot approach the solution with standard QP solvers. In our previous research [6] we derived only the binary formulation of the SND method while in the current paper we extend it to the multi-class case. In this setting one is interested in obtaining decision functions for each class respectively while trying to keep the data description compact [7]. This merges together objectives of novelty detection and classification and reveals the importance of bringing them together. The outliers in this scheme can be identified as the data which are not covered by any of the classes related to the obtained decision functions.

To illustrate the practical importance of the Supervised Novelty Detection we apply it to data from AVIRIS (Airborne Visible/InfraRed Imaging Sensor) [8]. Some previous papers on anomalous change detection [9], [10] already exploited the importance of SVM-based approaches in hyperspectral analysis of infrared images. However we can extend this along the lines of classification and detect hyperspectral changes among different types of terrain while trying to automatically categorize the pixels according to these types. Another promising application of SND are Intrusion Detection Systems (IDS). Here the goal is to identify intruders which might be scattered between many existing user groups. We cannot rely then on the fact that all users are originated from the same underlying distribution. Therefore many existing approaches would fail to generalize under the *i.i.d.* assumption. One might consider intruders as a separate class and resolve the problem in a multi-class fashion. But this approach is not very practical because of the initial diversity of intruders and high risk of overfitting of the resulting classifier. Combining One-Class with Multi-Class SVM might not be an optimal solution because of an added complexity and intermediate difficulties with integration in the provided solution.

The remainder of this paper is structured as follows. Section II gives a general view of our approach and discusses some related methods proposed in the literature. Section III gives some conventional notations and reviews the binary case of the SND method. Section IV outlines the multi-class QP and Least-Squares formulation while Section V extends the SND algorithm to large-scale problems with the newly derived optimization objective and provides theoretical bounds for convergence. Section VI discusses some implementation and algorithmic issues. Section VII provides the experimental setup and results. Finally Section VIII concludes the paper.

II. Problem statement and related work

A. Problem statement

Supervised Novelty Detection (SND) is designed for finding outliers in the presence of several classes/distributions. While

V. Jumutc and J.Suykens are with the Department of Electrical Engineering (ESAT-SCD-SISTA), Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Leuven, Belgium
E-mail: {Vilen.Jumutc, Johan.Suykens}@esat.kuleuven.be

being useful for detecting the outliers, the SND method can be effectively used for multi-class classification and it supplements the class of SVM-based algorithms. One can regard our approach as an extension of the original work by Schölkopf *et al.* [1] for One-Class SVM where one deals with the support of a high-dimensional distribution. Contrary to Schölkopf's approach we deal with labeled data and take the *i.i.d.* assumption for every class separately. We might also find some connections to [11] where the authors try to ablate outliers while trying to locate them with a new SVM objective reformulated in terms of a hinge loss. SND doesn't try to find outliers in the existing data pool of data. In general our objective is quite opposite. We try to find the support of each distribution per class such that we can identify outliers within our test or validation set while keeping a necessary discrimination between the observed classes. Moreover we can use outliers at the learning stage just by keeping their labels negative for all involved classes. This strategy helps to incorporate all available information at once.

B. Difference with other SVMs

We can think of SND as solving a density estimation problem for each involved distribution per class while trying to separate the classes as much as possible. In practice this results in finding an appropriate trade-off between the amount of errors, separation and compactness[1] of our model describing these particular distributions. The demonstrated problem is not of the same kind as other SVMs where one copes only with optimal separation (minimization of an average error) and the smoothness of the classifier. For instance, in Laplacian SVMs [12] one uses additional regularization to keep the values of the decision function for adjacent points similar but this regularization mostly affects unlabeled samples. In other methods [11] one is estimating outliers explicitly via a reformulated hinge-loss penalty. This setting is quite different from our objective of density estimation where we deal with the outliers either implicitly (see Section VI-B for further remarks) or explicitly by setting all respective labels to -1's.

III. Binary case

A. Notation

We first introduce terminology and some notational conventions. We consider training data with the corresponding labeling given as a set of pairs

$$(x_1, y_1), ..., (x_n, y_n), x_i \in \mathcal{X}, y_i \in \{-1, 1\},$$

where n is the number of corresponding observations in the set $\mathcal{X}$. Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^d$.

In Section III-C index i spans the range $\overline{1,n}$ if it is not declared explicitly. Greek letters $\alpha, \beta, \lambda, \xi$ without indices denote n-dimensional vectors, while in Section IV-A Greek letters $\alpha, \beta, \lambda, \xi$ spanning only one index denote n-dimensional vectors. In Section V letters w and x denote d-dimensional vectors. Otherwise Greek letters denote constants or scalars throughout the paper.

[1]by that we mean finding the smallest unit ball in the feature space that captures all the data, see [1] for details

B. Illustrative example

According to the classical work by Schölkopf *et al.* [1] in One-Class SVM we aim at mapping the data points into the feature space and separating them from the origin with maximum margin. From the joint perspective of density estimation for multiple distributions simultaneously we require more than only the compactness properties discussed in the previous section. From the model perspective we need a classification scheme which would preserve compactness and separation of distributions simultaneously. In our illustrative example we are

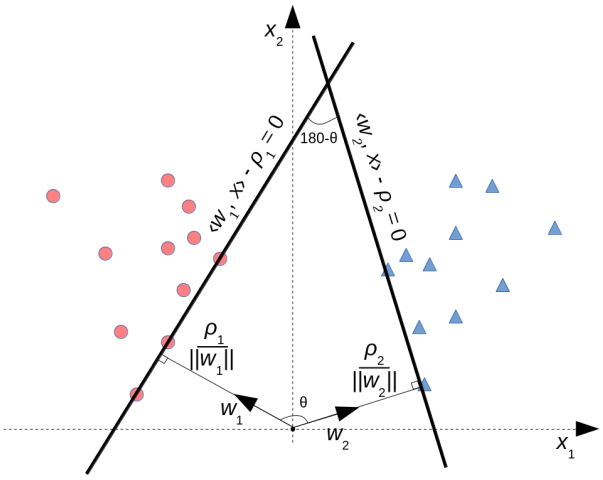


Fig. 1. SND solution in the feature space. SND aims at separating training data by minimizing the inner product between the normal vectors w_1 and w_2 to the decision hyperplanes while maximizing the margins (distances) between these hyperplanes and the origin.

emphasizing two core objectives of the SND method:

- maximizing margins $\frac{\rho_1}{\|w_1\|}$ and $\frac{\rho_2}{\|w_2\|}$,
- pushing θ closer to 180° angle (making $\cos\theta \simeq -1$).

If we take a look at the illustrative example in Figure 1 we can notice that these objectives are contradicting with each other. By making angle θ closer to 180 degrees we are making margins $\frac{\rho_1}{\|w_1\|}$ and $\frac{\rho_2}{\|w_2\|}$ smaller as it can be observed from Figure 2. This can be explained as well from the cosine perspective

$$\cos\theta = \frac{\langle w_1, w_2 \rangle}{\|w_1\|\|w_2\|}$$

as we should maximize $\|w_1\|$, $\|w_2\|$ (denominator) and minimize $\langle w_1, w_2 \rangle$ (numerator) in order to minimize the cosine and push angle θ closer to 180°. Following exactly this reasoning we present our binary QP problem in Section III where we trade-off the minimization of a coupling term $\langle w_1, w_2 \rangle$ in the cosine, minimization of the l_2-norms for the normal vectors w_1 and w_2 and the training errors ξ_i. We maximize the ρ_1, ρ_2 values as well as they do enter the definition of the margins for both decision hyperplanes.

In Figure 3 we show some clear advantages of the SND approach over One-Class SVM. The latter is not capable of identifying an outlier if it is located on the line connecting centroids of each distribution. One-Class SVM treats all samples as being drawn from the same distribution under the *i.i.d.* assumption.

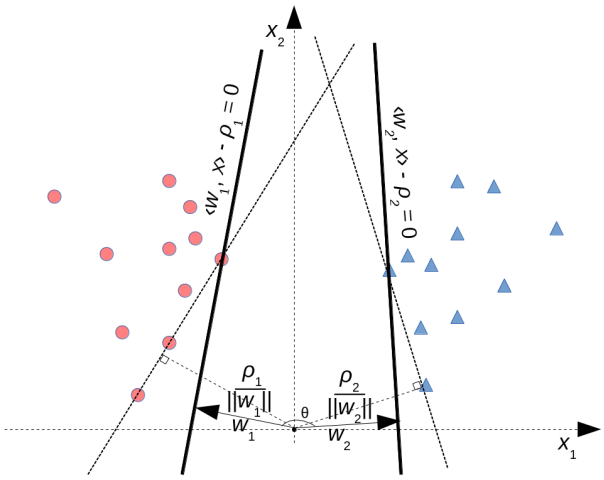


Fig. 2. SND solution in the feature space if we are emphasizing the second objective, making $\cos\theta \simeq -1$.

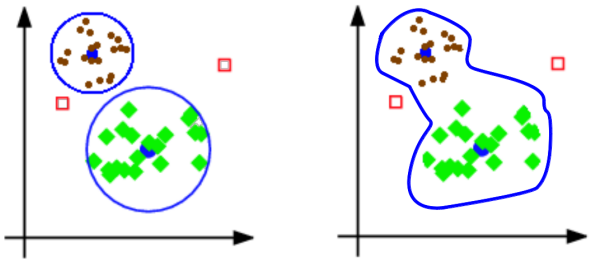


Fig. 3. Qualitative figure illustrating the main difference between SND solution (left) and One-Class SVM solution (right) in the input space. SND can provide the better and more compact estimate of each distribution. If an outlier sample (marked with the red square) was located on the line connecting centroids of each distribution One-Class SVM method would not detect such an outlier.

C. Binary QP problem

For the completeness we recap in this section the binary formulation of our approach [6] and then continue with the generalized multi-class QP and Least-Squares problem in the next sections.

First we start with the initial set of constraints which clarify the nature of our optimization problem w.r.t. normal vectors w_1, w_2 and maximization of the ρ bias terms [1], [13]

$$\begin{aligned}
&\langle w_1, \Phi(x_i)\rangle \geq \rho_1 - \xi_i^{(1)}, \quad \{x_i \in \mathcal{X} | y_i = 1\},\\
&\langle w_2, \Phi(x_i)\rangle \leq \rho_2 + \xi_i^{(2)}, \quad \{x_i \in \mathcal{X} | y_i = 1\},\\
&\langle w_1, \Phi(x_i)\rangle \leq \rho_1 + \xi_i^{(3)}, \quad \{x_i \in \mathcal{X} | y_i = -1\},\\
&\langle w_2, \Phi(x_i)\rangle \geq \rho_2 - \xi_i^{(4)}, \quad \{x_i \in \mathcal{X} | y_i = -1\},
\end{aligned} \tag{1}$$

where $y_i \in \{-1, 1\}$. To make a link between the One-Class SVM formulation and our method we join the constraints in Eq.(1) and propose the following optimization problem

$$\begin{aligned}
\min_{w_1,w_2\in\mathcal{F};\xi,\xi^*\in\mathbb{R}^n;\rho_1,\rho_2\in\mathbb{R}} \quad & \frac{\gamma}{2}(\|w_1\|^2 + \|w_2\|^2) + \langle w_1, w_2\rangle \\
& + C\sum_{i=1}^n(\xi_i + \xi_i^*) - \rho_1 - \rho_2
\end{aligned} \tag{2}$$

$$\begin{aligned}
\text{s.t.} \quad & y_i(\langle w_1, \Phi(x_i)\rangle - \rho_1) + \xi_i \geq 0, \quad i \in \overline{1,n}\\
& y_i(\langle w_2, \Phi(x_i)\rangle - \rho_2) - \xi_i^* \leq 0, \quad i \in \overline{1,n}\\
& \xi_i \geq 0, \xi_i^* \geq 0, \quad i \in \overline{1,n}
\end{aligned} \tag{3}$$

where γ and C are trade-off parameters. The decision functions are

$$\begin{aligned}
f_{c_1}(x) &= \langle w_1, \Phi(x)\rangle - \rho_1,\\
f_{c_2}(x) &= \langle w_2, \Phi(x)\rangle - \rho_2.
\end{aligned} \tag{4}$$

The final decision rule collects f_{c_1} and f_{c_2} as follows

$$c(x) = \begin{cases} \text{argmax}_{c_i}\ f_{c_i}(x), & \text{if } \max_i f_{c_i}(x) > 0\\ c_{out}, & \text{otherwise,}\end{cases} \tag{5}$$

where c_i is either the positive or negative class in the binary classification setting and c_{out} stands for the outliers' class.

Remark 1: Here we should stress the main difference with the binary classification setting where labels y_i are strongly associated with classes c_i. Our decision rule implies a separate class which doesn't directly enter the formulation in Eq.(2) but is thoroughly used for determining tuning parameters and calculation of the performance measures for our method. These data are assigned to an outliers' class as it doesn't belong to any of the encoded classes and can be seen as an unsupervised counterpart of our algorithm that can enter the optimization objective but those y_i labels for all classes will be set to -1. This is different from Laplacian SVMs [12] and manifold regularization [14]. The data $\mathcal{Z}$ are a subset of $\mathcal{X}$ defined as follows

$$z_1, \ldots, z_m \in \mathcal{Z} \subseteq \{\mathcal{X} : y_i = -1, i \in \overline{1, n_c}\}, \tag{6}$$

where n_c gives the total number of classes. This setting explicitly follows the multi-class case of Section IV and will be explained in detail in Section VI-B.

Using $\alpha_i, \lambda_i, \geq 0$ and $\beta_i, \beta_i^* \geq 0$ Lagrange multipliers we introduce the following Lagrangian

$$\begin{aligned}
\mathcal{L}(w_1, w_2, \xi, \xi^*, \rho_1, \rho_2, \alpha, \lambda, \beta, \beta^*) &= \tfrac{\gamma}{2}(\|w_1\|^2 + \|w_2\|^2)\\
&+\langle w_1, w_2\rangle + C\textstyle\sum_{i=1}^n(\xi_i + \xi_j^*)\\
&-\textstyle\sum_{i=1}^n \alpha_i(y_i(\langle w_1, \Phi(x_i)\rangle - \rho_1) + \xi_i)\\
&+\textstyle\sum_{i=1}^n \lambda_i(y_i(\langle w_2, \Phi(x_i)\rangle - \rho_2) - \xi_i^*)\\
&-\textstyle\sum_{i=1}^n \beta_i\xi_i - \sum_{i=1}^{n_c} \beta_i^*\xi_i^* - \rho_1 - \rho_2.
\end{aligned} \tag{7}$$

Before going to the final dual representation of Eq.(2) let Φ be a feature map $\mathcal{X} \to \mathcal{F}$ in connection to a positive definite Gaussian kernel [15], [16]

$$k(x, y) = \langle \Phi(x), \Phi(y)\rangle = e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{8}$$

By setting the derivatives of the Lagrangian with respect to the primal variables to zero, obtaining the saddle point conditions and substituting those into the Lagrangian one can directly obtain the matrix form of the corresponding Lagrangian to be maximized

$$\max_{\alpha,\lambda} \mathcal{L}_D(\alpha, \lambda) = \frac{\mu_1}{2}(\alpha^T G\alpha + \lambda^T G\lambda) + \mu_2(\alpha^T G\lambda), \tag{9}$$

$$\begin{aligned}
\text{s.t.} \quad & C \geq \alpha_i \geq 0, \ \forall i\\
& C \geq \lambda_i \geq 0, \ \forall i\\
& y^T\alpha = 1,\\
& y^T\lambda = -1,
\end{aligned} \tag{10}$$

where y is a vector of labels, K is the kernel matrix of dimension $n \times n$ with $K_{ij} = k(x_i, x_j) = \langle \Phi(x_i), \Phi(x_j)\rangle$,

$G = K \circ yy^T$, $\mu_1 = \frac{\gamma}{1-\gamma^2}$, $\mu_2 = \frac{1}{1-\gamma^2}$, and $\circ$ denotes component-wise multiplication. $\mathcal{L}_D$ is maximized and supplements the class of QP problems with box constraints. We can ensure the concavity of our dual objective in Eq.(9) by setting $\gamma > 1$. The latter condition is a straightforward consequence from the eigendecomposition of the matrix in the quadratic form of our optimization objective.

IV. Multi-class case

A. Multi-class QP problem

In this subsection we develop a generic QP formulation for the multi-class setting of our algorithm which returns decision functions f_i for each of the involved target classes (distributions). These functions encode the support for each distribution and output positive values in a corresponding region capturing most of the data points drawn from it.

Combining ideas from One-Class SVM and our assumption we presented previously in Section III-C the following QP problem is formulated

$$\min_{w_i \in \mathcal{F}; \xi_i \in \mathbb{R}^n; \rho_i \in \mathbb{R}} \quad \frac{\gamma}{2}\sum_{i=1}^{n_c}\|w_i\|^2 + \sum_{i,j=1;i\neq j}^{n_c}\langle w_i, w_j\rangle + C\sum_{i=1}^{n}\sum_{j=1}^{n_c}\xi_{ij} - \sum_{i=1}^{n_c}\rho_i \tag{11}$$

$$\text{s.t.} \quad y_{ij}\langle w_j, \Phi(x_i)\rangle \geq \rho_j - \xi_{ij}, \quad i \in \overline{1,n}, \quad j \in \overline{1,n_c}$$
$$\xi_{ij} \geq 0, \qquad\qquad\quad i \in \overline{1,n}, \quad j \in \overline{1,n_c} \tag{12}$$

where $y_{ij} \in \{-1, 1\}$, γ and C are trade-off parameters and n_c is the number of classes. Here we observe that we are working with the set of indices $\mathcal{Y}$, where every entry $y_i \in \{-1, 1\}^{n_c}$. The decision functions are

$$f_{c_i}(x) = \langle w_i, \Phi(x)\rangle - \rho_i, \tag{13}$$

and the final decision rule is derived in Eq.(5). Using $\alpha_{ij}, \beta_{ij} \geq 0$ as Lagrange multipliers we introduce the following Lagrangian

$$\mathcal{L}(w, \xi, \rho, \alpha, \beta) = \frac{\gamma}{2}\sum_{i=1}^{n_c}\|w_i\|^2 + \sum_{i,j=1;i\neq j}^{n_c}\langle w_i, w_j\rangle + C\sum_{i=1}^{n}\sum_{j=1}^{n_c}\xi_{ij} - \sum_{i=1}^{n}\rho_i - \sum_{i=1}^{n}\sum_{j=1}^{n_c}\beta_{ij}\xi_{ij} - \sum_{i=1}^{n}\sum_{j=1}^{n_c}\alpha_{ij}(y_{ij}\langle w_j, \Phi(x_i)\rangle - \rho_j + \xi_{ij}). \tag{14}$$

By setting the derivatives of the Lagrangian with respect to the primal variables to zero and defining $\eta = \gamma + n - 2$ we obtain

$$w_i = \frac{\eta\sum_{j=1}^{n}\alpha_{ji}y_{ji}\Phi(x_j) - \sum_{j=1}^{n}\sum_{p=1,p\neq i}^{n_c}\alpha_{jp}y_{jp}\Phi(x_j)}{(\eta+1)(\gamma-1)}, \tag{15}$$

$$C - \beta_{ij} - \alpha_{ij} = 0, \quad \forall i \in \overline{1,n} \quad \forall j \in \overline{1,n_c} \tag{16}$$

$$\sum_{i=1}^{n}\alpha_{ij} = 1, \quad \forall j \in \overline{1,n_c}. \tag{17}$$

Substituting Eq.(15-17) into the Lagrangian and using the kernel trick with the expression given by Eq.(8) one can directly obtain the matrix form of the corresponding Lagrangian to be maximized

$$\max_{\alpha_i} \mathcal{L}_D(\alpha_i) = \frac{1}{\mu}\sum_{i}^{n_c}\lambda_i^T K\alpha_i, \tag{18}$$

$$\text{s.t.} \quad C \geq \alpha_{ij} \geq 0, \ \forall i \in \overline{1,n}, \ \forall j \in \overline{1,n_c}$$
$$\sum_{i=1}^{n}\alpha_{ij} = 1, \ \forall j \in \overline{1,n_c} \tag{19}$$

where $\lambda_i = (\gamma + n - 2)(\alpha_i \circ y_i) - \sum_{j=1,j\neq i}^{n_c}(\alpha_j \circ y_j)$, $\mu = (\eta+1)(\gamma-1)$, K is a kernel matrix of size $n \times n$ and $\circ$ denotes component-wise multiplication. $\mathcal{L}_D$ is maximized and is almost identical to one defined in Eq.(9) if we take $n_c = 2$. The expression for f_i becomes

$$f_{c_i}(x) = \frac{\eta\sum_{j=1}^{n}\alpha_{ji}y_{ji}k(x_j, x) - \sum_{j}^{n}\sum_{p=1,p\neq i}^{n_c}\alpha_{jp}y_{jp}k(x_j, x)}{(\eta+1)(\gamma-1)} - \rho_i, \tag{20}$$

where $k(x, y)$ stands for our preferred kernel function in Eq.(8).

We can ensure the concavity of our dual objective in Eq.(18) by examining necessary conditions for the primal problem in Eq.(11) to be strictly convex. This can be done by applying the Gershgorin circle theorem to bound the minimal positive eigenvalue. It is very easy to verify when $\gamma > n_c - 1$ we have $\lambda_{min} > 0$.

B. Least-Squares problem

To obtain Least-Squares (LS-SND) formulation with equality constraints of our initial problem we reformulate Eq.(11) in terms of squared error residuals ξ_{ij}

$$\min_{w_i \in \mathcal{F}; \xi_i \in \mathbb{R}^n; \rho_i \in \mathbb{R}} \quad \frac{\gamma_1}{2}\sum_{i=1}^{n_c}\|w_i\|^2 + \sum_{i,j=1;i\neq j}^{n_c}\langle w_i, w_j\rangle + \frac{\gamma_2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n_c}\xi_{ij}^2 - \sum_{i=1}^{n_c}\rho_i \tag{21}$$

$$\text{s.t.} \quad y_{ij}\langle w_j, \Phi(x_i)\rangle = \rho_j - \xi_{ij}, \quad i \in \overline{1,n}, \quad j \in \overline{1,n_c}. \tag{22}$$

The Lagrangian for this problem is

$$\mathcal{L}(w_i, \xi, \rho, \alpha) = \frac{\gamma_1}{2}\sum_{i=1}^{n_c}\|w_i\|^2 + \sum_{i,j=1;i\neq j}^{n_c}\langle w_i, w_j\rangle + \frac{\gamma_2}{2}\sum_{i=1}^{n}\sum_{j=1}^{n_c}\xi_{ij}^2 - \sum_{i=1}^{n_c}\rho_i - \sum_{i=1}^{n}\sum_{j=1}^{n_c}\alpha_{ij}(y_{ij}\langle w_j, \Phi(x_i)\rangle - \rho_j + \xi_{ij}), \tag{23}$$

where the α_{ij} values are the Lagrange multipliers which can be both positive and negative now due to the equality constraints.

By substituting $\eta = \gamma_1 + n - 2$ the conditions for optimality now yield

$$w_i = \frac{\eta\sum_{j}^{n}\alpha_{ji}y_{ji}\Phi(x_j) - \sum_{j}^{n}\sum_{p=1,p\neq i}^{n_c}\alpha_{jp}y_{jp}\Phi(x_j)}{(\eta+1)(\gamma_1-1)}, \tag{24}$$

$$\alpha_{ij} = \gamma_2\xi_{ij}, \quad \forall i \in \overline{1,n} \quad \forall j \in \overline{1,n_c} \tag{25}$$

$$\sum_{i=1}^{n}\alpha_{ij} = 1, \quad \forall j \in \overline{1,n_c}. \tag{26}$$

By substituting the expressions for w_i and ξ_{ij} in our equality condition, applying the kernel trick in Eq.(8) and preserving matrices $G_{ij} = K \circ y_i y_j^T$ and constants from Eq.(19) we can obtain the following linear Karush-Kuhn-Tucker (KKT) system of the form

$$\Omega\alpha^\star = \theta, \tag{27}$$

which we solve in α_i and ρ_i, where

$$\Omega = \left(\begin{array}{c|c} \begin{matrix} 0 \dots 0 \\ \vdots \ddots \vdots \\ 0 \dots 0 \end{matrix} & \begin{matrix} 1_1^T \dots 0_n^T \\ \vdots \ddots \vdots \\ 0_n^T \dots 1_n^T \end{matrix} \\ \hline \begin{matrix} \\ 1_n \dots 0_n \\ \vdots \ddots \vdots \\ 0_n \dots 1_n \end{matrix} & \begin{matrix} -\frac{\eta G^\star_{11}}{\mu} & \dots & \frac{G_{1n_c}}{\mu} \\ \vdots & \ddots & \vdots \\ \frac{G_{n_c1}}{\mu} & \dots & -\frac{\eta G^\star_{n_cn_c}}{\mu} \end{matrix} \end{array}\right) \tag{28}$$

defining $G^\star_{ij} = G_{ij} + \frac{\mu}{\eta\gamma_2} I$ and

$$\alpha^\star = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_{n_c} \\ \hline \alpha_1 \\ \vdots \\ \alpha_{n_c} \end{pmatrix} \qquad \theta = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ \hline 0_n \\ \vdots \\ 0_n \end{pmatrix} \tag{29}$$

and 1_n and 0_n denote vectors of length n. To clarify the structure of the matrix Ω we should refer to every part of this matrix separately. The upper-left submatrix is a square matrix of size $n_c \times n_c$ where all residuals are zeros. The upper-right and bottom-left matrices are block diagonal where every element on the diagonal is a vector 1_n. These matrices are identical but the upper-right matrix is transposed. The bottom-right matrix is a square matrix of size $nn_c \times nn_c$ where every element on the diagonal is of the form $-\frac{\eta}{\mu}(G_{ii}+I/\gamma_2)$ and any off-diagonal element is bound to matrix G_{ij} in the following form: $\frac{G_{ij}}{\mu}$. The final decision function and the decision rule are of the same form as in Eq.(20) and Eq.(5).

Remark 2: Additionally we should emphasize that the Least-Squares form of our algorithm is of much less complexity than QP formulation and results in only one linear system of size $nn_c \times nn_c$. This drastically decreases computational costs for the cross-validation procedure which will be presented in Section VII-A and mentioned in the description of Algorithms 2 – 3.

V. Large-scale optimization problem

A. Algorithm

To cope with large-scale datasets we propose a scalable first-order optimization algorithm for the multi-class QP problem. The formulation is inspired by the Pegasos algorithm [17] and we provide theoretical justification along the lines of the Pegasos formulation.

Remark 3: The large amount of variables significantly slows down every iteration of any QP solver and starting from several thousands of variables even our approach for tuning the parameters (see Section VII-A) becomes unfeasible. To tackle this problem one may study a scalable SMO-like method by Platt [18] or Nesterov's approach for convex optimization [19]. However we selected here a Pegasos-like implementation of the SND algorithm which makes use of the Nyström approximation of the RBF kernel [20], [21] and converges with the selected accuracy ϵ within $\mathcal{O}(\frac{R^2}{\lambda\epsilon})$ iterations. This result originally provided in [17] is much better than previously implemented approaches (e.g. SVM-Perf [22]) which like Pegasos make use of the subgradient descent but converge in $\mathcal{O}(\frac{R^2}{\lambda\epsilon^2})$.

First we rewrite our optimization objective in Eq.(11) in terms of the hinge loss. Second we move the bias terms ρ_i into the hinge loss. Finally we optimize only over the weights w_i which are joint together as

$$w = \begin{pmatrix} w_1 \\ \vdots \\ w_{n_c} \end{pmatrix}$$

to be compatible with the original formulation of the Pegasos algorithm. We benefit from the convergence analysis provided in [17] and present our adjustments for the SND method in Theorem 1.

We derive an approximate instantaneous objective function in the primal for the SND method by

$$f(w; \mathcal{A}_t; B_i; \Gamma) = \frac{\lambda}{2} w^T \Gamma w + \frac{1}{m} \sum_{i=1}^{n_c} \sum_{(x,y)\in\mathcal{A}_t} \mathbb{L}(w; B_i; (x, y)), \tag{30}$$

where the hinge loss for the i-th class is given by

$$\mathbb{L}(w; B_i; (x, y)) = \max\{0, 1 - y(\langle w, B_i^T x\rangle + \rho_i)\}, \tag{31}$$

and $\mathcal{A}_t$ is our working subset (subsample) at iteration t and matrices Γ and B_i are of the special form

$$\Gamma = \begin{pmatrix} \gamma I_{11} & \dots & I_{1n_c} \\ \vdots & \ddots & \vdots \\ I_{n_c1} & \dots & \gamma I_{n_cn_c} \end{pmatrix}, \tag{32}$$

$$B_i = \begin{pmatrix} 0 & \dots & I_i & \dots & 0 \end{pmatrix}.$$

In the above equations we expect w to be of dimension dn_c where d is our input dimension and n_c is the number of classes. Every identity matrix or zero matrix is of dimension $d \times d$ and $\rho_i \in \mathbb{R}$. Scalar m denotes the size of the working subset $\mathcal{A}_t$.

Here we should emphasize that we carry out optimization only *w.r.t.* w and we include ρ (which is part of the hinge loss) as a additional (last) element of vector w. This strategy, originally proposed in [17], allows us to rely on the strong convexity of the optimization objective.

Next we present a brief summary of the large-scale SND method in Algorithm 1 and continue with the analysis in the next subsection. Below we denote the whole dataset by $\mathcal{S}$.

The above algorithm is based on the Pegasos formulation but differs in the computation of the subgradient and the projection step. Now we can see that the subgradient

$$\nabla_t = \lambda\Gamma w^{(t)} - \frac{1}{m} \sum_{i=1}^{n_c} \sum_{(x,y_i)\in\mathcal{A}^+_{t(i)}} y_i B_i^T x \tag{33}$$

Algorithm 1: Pegasos-based SND algorithm

Data: $\mathcal{S}, \gamma, \lambda, T, m$

Compute Γ and B_i matrices defined in Eq.(32)

Set $w^{(1)}$ randomly s.t. $\|w^{(1)}\| \leq \sqrt{n_c/\lambda(\gamma + n_c - 1)}$

for $t = 1 \rightarrow T$ **do**

  Set $\eta_t = \frac{1}{\lambda t}$

  Select $\mathcal{A}_t \subseteq \mathcal{S}$, where $|\mathcal{A}_t| = m$

  $\mathcal{A}^+_{t(i)} = \{(x, y) \in \mathcal{A}_t : y(\langle w, B_i^T x\rangle) < 1\}, \forall i$

  $w^{(t+\frac{1}{2})} = w^{(t)} - \eta_t(\lambda\Gamma w^{(t)} - \frac{1}{m}\sum_{i=1}^{n_c}\sum_{(x,y)\in\mathcal{A}^+_{t(i)}} yB_i^T x)$

  $w^{(t+1)} = \min\left\{1, \frac{\sqrt{n_c/\lambda(\gamma+n_c-1)}}{\|w^{(t+\frac{1}{2})}\|}\right\} w^{(t+\frac{1}{2})}$

end

return $w^{(T+1)}$

depends on the additional matrices Γ and B_i introduced in Eq.(32) and in projection step (10) we have slightly different rescaling term.

B. Analysis

In this subsection we present a convergence analysis which brings to our algorithm the same convergence bounds as in Pegasos. We extend the analysis presented in [17] to our instantaneous objective by presenting Theorem 1. But first we recap the important lemma from [17] which establishes necessary conditions for our theorem.

Lemma 1 (Shalev-Shwartz et al., 2007): Let $f^{(1)}, ..., f^{(T)}$ be a sequence of λ-strongly convex functions w.r.t. the function $\frac{1}{2}\|\cdot\|^2$. Let $\mathcal{B}$ be a closed convex set and define $\prod_{\mathcal{B}}(w) = \arg\min_{w'\in\mathcal{B}}\|w - w'\|$. Let $w^{(1)}, \ldots, w^{(T+1)}$ be a sequence of vectors such that $w^{(1)} \in \mathcal{B}$ and for $t \geq 1$, $w^{(t+1)} = \prod_{\mathcal{B}}(w^{(t)} - \eta_t\nabla_t)$, where ∇_t is a subgradient of $f^{(t)}$ at $w^{(t)}$ and $\eta_t = 1/\lambda t$. Assume that for all t, $\|\nabla_t\| \leq G$. Then, for all $u \in \mathcal{B}$ we have

$$\frac{1}{T}\sum_{t=1}^{T} f(w^{(t)}) \leq \frac{1}{T}\sum_{t=1}^{T} f(u) + \frac{G^2(1 + \ln(T))}{2\lambda T}.$$

Based on the above lemma, we are now ready to bound the average instantaneous objective of Algorithm 1.

Theorem 1: Assume $\|x\| \leq R$ for all $(x, y) \in \mathcal{S}$. Let $w^* = \arg\min_w f(w; \mathcal{A}_t; B_i; \Gamma)$ and let $c = \sqrt{\lambda n_c(\gamma + n_c - 1)} + n_c R$. Then, for $T \geq 3$ and $\gamma > n_c - 1$ we have

$$\frac{1}{T}\sum_{t=1}^{T} f(w^{(t)}; \mathcal{A}_t; B_i; \Gamma) \leq \frac{1}{T}\sum_{t=1}^{T} f(w^*; \mathcal{A}_t; B_i; \Gamma) + \frac{c^2\ln(T)}{\lambda T}.$$

Proof: To prove our theorem it suffices to show that all conditions of Lemma 1 hold. First we show that our problem is strongly convex. It is easy to verify that matrix Γ given in Eq.(32) is always positive definite if $\gamma > n_c - 1$ which implies that Bregman divergence is always bounded from below w.r.t to λ and 2-norm $\|\cdot\|$. Since $f^{(t)}$ is a sum of λ-strongly convex function $\frac{\lambda}{2}w^T\Gamma w$ and another convex function (hinge-loss), it is also λ-strongly convex. Next by assuming $\mathcal{B} = \{w : \|w\| \leq \sqrt{n_c/\lambda(\gamma + n_c - 1)}\}$ and the fact that $\|x\| \leq R$ we can bound subgradient ∇_t. The explicit form for the subgradient evaluated at point x is given in Eq.(33). Using the triangular inequality and denoting 2-norm by $\|\cdot\|$ one obtains

$$\|\nabla_t\| \leq \lambda\|\Gamma w\| + \textstyle\sum_i \|B_i^T x\| \leq \lambda\|\Gamma\|\|w\| + n_c\|x\| \leq$$
$$\leq \lambda(\gamma + n_c - 1)\|w\| + n_c R \leq \sqrt{\lambda n_c(\gamma + n_c - 1)} + n_c R.$$

The upper bound on $\|\Gamma\|$ is derived using the Gershgorin circle theorem as follows:

$$\|\Gamma\| \leq \sqrt{\upsilon_{max}(\Gamma^*\Gamma)} = \upsilon_{max}(\Gamma) \leq D(\gamma, n_c - 1) = \gamma + n_c - 1,$$

where Γ^* is the conjugate transpose of Γ, υ_{max} is the maximum eigenvalue and $D(\gamma, n_c - 1)$ is the Gershgorin circle with the center γ and radius $n_c - 1$. The first equality follows from the block-wise structure of matrix Γ. The last inequality follows from the fact that diagonal elements of Γ are the same and equal to γ everywhere and the sum of off-diagonal elements is exactly $n_c - 1$, which is clear from the structure of Γ in Eq.(32). Finally we have to show that $w^* \in \mathcal{B}$. To do so, we derive the dual form of our objective in terms of the dual variables $\alpha_i \in [0, 1]^n, i \in \overline{1, n_c}$ related to decision functions f_{c_i} in Eq.(13) such that we have the following mixed optimization objective

$$\max_{\alpha_i}\min_{w} \frac{1}{m}\sum_{i=1}^{n_c}\|\alpha_i\|_1 - \frac{\lambda}{2}w^T\Gamma w$$

and after assuming strong duality and the optimal solution *w.r.t* the primal variable w^* and dual variables α_i^* one gets

$$\frac{\lambda}{2}w^{*T}\Gamma w^* + \frac{1}{m}\sum_{i=1}^{n_c}\sum_{x\in\mathcal{S}}\mathbb{L}(w^*; x) = -\frac{\lambda}{2}w^{*T}\Gamma w^* + \frac{1}{m}\sum_{i=1}^{n_c}\|\alpha_i^*\|_1.$$

For simplicity we replace the notation for the hinge-loss with $\mathbb{L}(w^*; x)$. Rearranging the above, using the non-negativity of the hinge-loss and applying the Gershgorin circle theorem we obtain our bound: $\|w\| \leq \sqrt{n_c/\lambda(\gamma + n_c - 1)}$. Now we can plug-in everything back to inequality in Lemma 1 which completes the proof. ■

C. Fixed-Size approach

One of the crucial aspects in estimating the support of some unknown high-dimensional distribution is the non-linearity of the feature space where we are trying to find a solution. As it was discussed in [1] we cannot rely on the linear kernel in this case and should use the RBF kernel instead. To overcome restrictions of Algorithm 1 which operates only in the primal space we apply a Fixed-Size approach [20] to approximate the RBF kernel with some higher dimensional explicit feature vector.

First we use an entropy based criterion to select the prototype vectors (small working sample of size $m \ll n$)[2] and construct kernel matrix K. Based on the Nyström approximation [21] an expression for the entries of the approximation of the feature map $\hat{\Phi}(x) : \mathbb{R}^d \rightarrow \mathbb{R}^m$, with $\hat{\Phi}(x) = (\hat{\Phi}_1(x), \ldots, \hat{\Phi}_m(x))^T$ is given by

$$\hat{\Phi}_i(x) = \frac{1}{\sqrt{\lambda_{i,m}}}\sum_{t=1}^{m} u_{ti,m}k(x_t, x),$$

[2] see Section 4 of [20] for the details

where $\lambda_{i,m}$ and $u_{i,m}$ denote the i-th eigenvalue and the i-th eigenvector of K defined in Eq.(8). Using the above expression for $\hat{\Phi}(x)$ we can proceed with the original formulation of Algorithm 1 and find the solution of our problem in primal.

VI. Algorithms and explanations

A. Coupling term and γ explained

To illustrate the importance of the coupling term $\langle w_i, w_j \rangle$ we implemented a toy example where initially the coefficient γ in Eq.(11, 21) is fixed and the other hyperparameters were obtained via the tuning procedure described in Section VII-A.

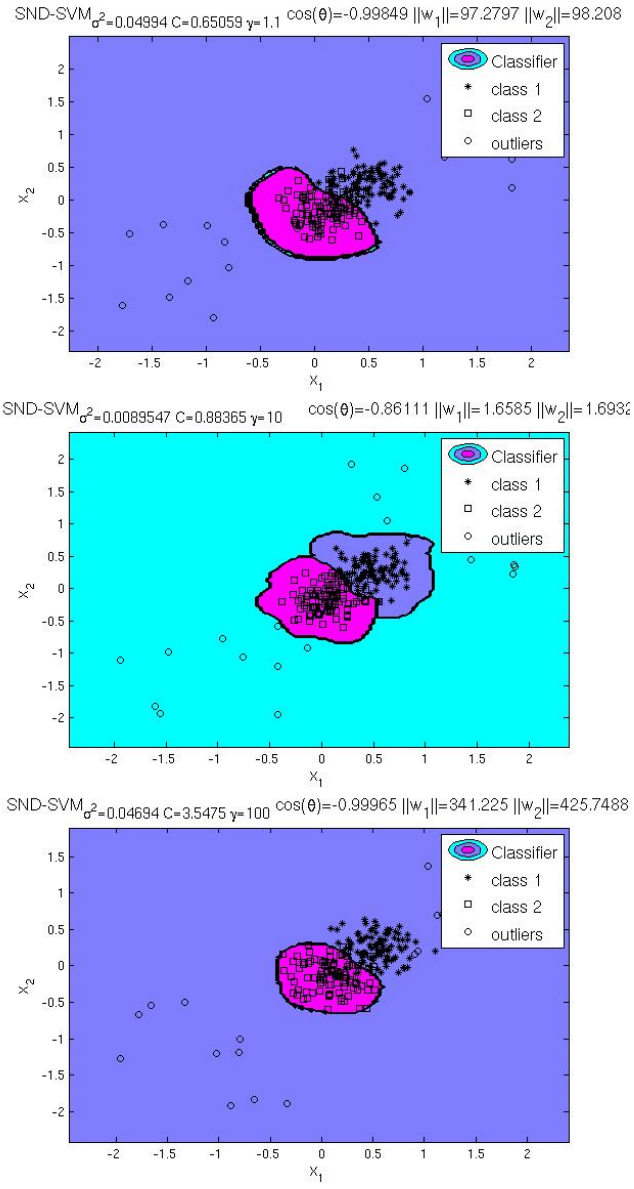


Fig. 4. Decision boundaries of the SND method for varying values of the γ hyperparameter, illustrating the importance of small $cos\theta$ and minimized $\|w_1\|, \|w_2\|$.

As we can see in Figure 4 the parameter γ directly affects the decision boundaries of the SND method as it increases from 1.1 in the topmost subfigure to 100 in the bottom one. To facilitate the reasoning of how γ value affects the coupling term and and the overall model consistency we provide each subfigure with the effective value of $\|w_1\|$, $\|w_2\|$ and $\cos\theta$ terms which are calculated *w.r.t.* our dual representation in Eq.(9) and the kernel expansion in Eq.(8) as

$$\cos\theta = \frac{\langle w_1, w_2 \rangle}{\|w_1\|\|w_2\|} = \frac{\alpha^T G \lambda}{\sqrt{(\alpha^T G \alpha)(\lambda^T G \lambda)}},$$

where $\|w_1\| = \sqrt{\alpha^T G \alpha}$, $\|w_2\| = \sqrt{\lambda^T G \lambda}$ and $G = K \circ yy^T$ relates to the matrix calculated from the training data. From examining Figure 4 one can observe that only carefully chosen parameter γ and a trade-off for $\langle w_1, w_2 \rangle$ term can bring necessary discrimination between classes while preserving the compactness of the support. This means that any over- or underestimation of γ parameter can lead to an unsatisfactory solution. The central subfigure of Figure 4 clearly indicates that a minimal $\cos\theta$ term doesn't ensure the best possible solution. This fact empirically illustrates our intuition and reasoning about the relation between the coupling term and margins as the top and bottom subfigures provide a good separation between classes but do not ensure the compact support for one of the distributions. We can see that $\|w_1\|$, $\|w_2\|$ are quite large (of 10^2 magnitude) and one of the classes almost completely covers the entire space.

B. Classification and novelty detection algorithms

In this section we present a general purpose algorithm for SND which can be applied both in classification and novelty detection settings.

To clarify how the SND method can be used in both settings: classification and novelty detection, we present a brief algorithmic summary for these settings in Algorithms 2–3. One should notice that the main difference between both algorithms is the cross-validation step, decision rule and the input data.

In the presented algorithms the "CrossvalidateSND" function stands for the tuning procedure which will be described in the next section. The crucial difference between Algorithm 2 and 3 is the usage of the data Z defined in Eq.(6). The SND model is tuned to perform novelty detection with respect to data Z and maximize the observed detection rate. In binary classification problem in Eq.(2) we cannot use data Z because of the labeling limitation on $y_i \in \{-1, 1\}$. We have to switch to the multi-class optimization objective in Eq.(11). Here we refer to Z as a matrix containing subset $\mathcal{Z} \subseteq \mathcal{X}$ which is labeled negatively everywhere, by taking $y_i = -1, i \in \overline{1, n_c}$. It can be used in the cross-validation procedure, such that we do care about maximizing detection rate of those samples along with minimization of the validation error for positively labeled samples. As a result of the "CrossvalidateSND" function we output the optimal parameters γ, C for the SND model and the optimal RBF kernel width σ. Finally $c(x)$ decision functions are defined by the means of the dual variables α_i, the primal variables ρ_i, the optimal parameters γ, σ and the labeling Y in Eq.(5) and Eq.(20). Here we can notice that for Algorithm 2 we are not giving any alternative decisions in $c(x)$ and are obliged to select between classes c_i.

Algorithm 2: SND for binary classification

input : training data X of size $l \times d$, class labels Y of size $l \times n_c$

output: SND explicit decision rule

begin

$[\gamma, \sigma, C] \leftarrow \texttt{CrossvalidateSND}\,(X, Y)$;

$[\alpha, \rho] \leftarrow \texttt{ComputeSND}\,(X, Y, \gamma, \sigma, C)$;

$c(x) \leftarrow \text{argmax}_{c_i}\ f_{c_i}(x)$;

end

Algorithm 3: SND for novelty detection

input : training data X of size $l \times d$, outliers' data Z of size $m \times d$, class labels Y of size $l \times n_c$, -1_z matrix of minus ones of size $m \times n_c$

output: SND explicit decision rule

begin

$[\gamma, \sigma, C] \leftarrow \texttt{CrossvalidateSND}\,(X, Y, Z, -1_z)$;

$[\alpha, \rho] \leftarrow \texttt{ComputeSND}\,([X; Z], [Y; -1_z], \gamma, \sigma, C)$;

$c(x) \leftarrow \begin{cases} \text{argmax}_{c_i}\ f_{c_i}(x), & \text{if } \max_i f_{c_i}(x) > 0 \\ c_{out}, & \text{otherwise} \end{cases}$;

end

VII. Empirical results

A. Experimental setup

In all our experiments for all tested SND and SVM models we use a 2-step procedure for tuning the parameters. This procedure consists of Coupled Simulated Annealing [23] initialized with 5 random sets of parameters for the first step and the simplex method [24] for the second step. After CSA converges to some local minima we select the tuple of parameters that attains the lowest error and start the simplex procedure to refine our selection. On every iteration step for CSA and simplex method we proceed with a 10-fold cross-validation. While being considerably faster than the straightforward grid search technique obtained parameters tend to vary more because of the randomness in initialization.

We selected the universal RBF kernel (see [25]) that is generally capable to separate all compact subsets and is suitable for many kinds of data. The choice of the RBF kernel was motivated by [1] where the authors explain an obvious advantage of it and that the data are always separable from the origin in the feature space (see Definition 1 in [1]). We tune the bandwidth of the RBF kernel in Eq.(8) with additional trade-off parameters for all methods using the tuning procedure described within the previous paragraph.

For the large-scale version of SND we use the Nyström approximation and the Fixed-Size approach [20] where the σ parameter was inferred via cross-validation procedure described above. The active subset was selected via maximization of the Renyi entropy. The size of this subset was set to be $\sqrt{n}$ for all methods that utilize Nyström approximation. Finally we fix the m parameter in Algorithm 1 to be $0.1|\mathcal{S}|$.

For the Toy Data (1) we performed 100 iterations with random sampling of size 100 according to the separate uniform distributions from intersecting intervals $[0, 1]$ and $[-0.5, 0.5]$, collected averaged error rates with corresponding standard deviations. For novelty detection we performed 100 iterations with random sampling from three different distributions[3] (see Figure 6) scaled to the range $[-1, 1]$ for all dimensions. For all toy datasets in every iteration we splitted all data points in proportion 80% to 20% into training and test counterparts. In novelty detection setting 15% of all data samples were generated as outliers. For all UCI datasets [26] (except for Arcene and large scale datasets) we used 5 independent 10-fold splittings and performed averaging and paired t-tests [27] for the comparison of errors. Arcene was split into training and validation datasets initially and we simply run the classification scheme 10 times. For the large scale datasets we run all methods 50 times with the random split in proportion of 50% by 50% for training and test data respectively. For the properties of UCI and toy datasets one can refer to the Table I.

TABLE I
Datasets

Dataset	# of attributes	# of classes	# of data points
Toy Data (1)	2	2	200
Toy Data (2-4)	2	2	150
Arcene	10000	2	900
Ionosphere	34	2	351
Parkinsons	23	2	197
Sonar	60	2	208
Zoo	17	7	101
Iris	4	3	150
Ecoli	8	5	336
TAE	5	3	151
Seeds	7	3	210
Arrhythmia	279	2	452
Pima	8	2	768
Madelon	500	2	2000
Red Wine	12	2	1599
White Wine	12	2	4898
Magic	11	2	19020

We implemented the original QP formulation of the SND method as an optimization problem using the Ipopt package (see [28]), which implements a general purpose interior point search algorithm. The Least-Squares version of SND was implemented using standard Matlab backslash operation. The large-scale version of SND and Pegasos were implemented in Matlab. LS-SVM with Fixed-Size approach is entirely implemented in Matlab as well. For learning C-SVM and One-Class SVM we used the LIBSVM package [29]. All experiments were run on Core i7 CPU with 8GB of RAM available under Linux CentOS platform.

B. Numerical results with UCI datasets

First we present some results for the classification setting where we can fairly compare our method to C-SVM [15] and LS-SVM [30]. Then we proceed with some results for the large-scale UCI datasets. Then we continue with the novelty detection scheme in the presence of two and more classes and some number of outliers. Here we simply present preliminary

[3]Toy Data (2-4)

results for different toy problems and report performance in terms of general test error and detection rate[4]. Finally in the next subsection we present real life example from anomalous change detection in AVIRIS (Airborne Visible/InfraRed Imaging Sensor) images [8].

TABLE II
AVERAGED MISCLASSIFICATION ERROR ON TEST DATA

Dataset	SND	C-SVM	LS-SVM
Toy Data (1)	0.1395± 0.097	0.1385± 0.078	**0.1325**± 0.085
Arcene	**0.1620**± 0.006	0.1730± 0.095	0.1810± 0.091
Ionosphere	0.0684± 0.043	0.0740± 0.031	**0.0483**± 0.030
Parkinsons	**0.0613**± 0.046	0.0721± 0.060	0.0621± 0.064
Sonar	**0.0962**± 0.069	0.1250± 0.105	0.1205± 0.101
Zoo	**0.0500**± 0.081	0.0733± 0.119	0.1071± 0.119
Iris	0.0467± 0.068	**0.0440**± 0.065	0.0493± 0.067
Ecoli	0.1263± 0.069	**0.1240**± 0.061	0.1562± 0.062
TAE	**0.4031**± 0.159	0.4346± 0.146	0.5545± 0.131
Seeds	0.0667± 0.060	**0.0650**± 0.050	0.0838± 0.073

TABLE III
AVERAGED MISCLASSIFICATION ERROR ON TEST DATA

Dataset	LS-SND	C-SVM	LS-SVM
Toy Data (1)	0.1425± 0.079	0.1450± 0.081	**0.1395**± 0.079
Ionosphere	0.0803± 0.033	0.0705± 0.044	**0.0541**± 0.034
Parkinsons	**0.0566**± 0.046	0.0664± 0.065	0.0647± 0.050
Sonar	0.1198± 0.059	**0.1173**± 0.074	0.1283± 0.054
Arrhythmia	**0.2193**± 0.050	0.2220± 0.050	0.2286± 0.061
Pima	0.2325± 0.039	**0.2308**± 0.043	0.2391± 0.039
Zoo	0.1487± 0.145	**0.0671**± 0.079	0.1518± 0.109
Iris	0.0667± 0.070	0.0427± 0.060	**0.0347**± 0.043
Ecoli	0.1586± 0.084	**0.1192**± 0.044	0.1376± 0.040
TAE	**0.4219**± 0.110	0.4300± 0.141	0.5655± 0.116
Seeds	0.0905± 0.063	**0.0629**± 0.049	0.0905± 0.063

TABLE IV
AVERAGED MISCLASSIFICATION ERROR ON TEST DATA

Dataset	SND	Pegasos	NyFS-LSSVM
Pima	0.2885± 0.024	0.2866± 0.020	0.2333± 0.020
Madelon	0.4307± 0.022	0.4272± 0.017	0.4531± 0.014
Red Wine	0.2648± 0.016	0.2625± 0.014	0.2583± 0.014
White Wine	0.2747± 0.021	0.2715± 0.014	0.2381± 0.008
Magic	0.1474± 0.012	0.1576± 0.004	0.1375± 0.003

Tables II-III present results for independent runs of QP and Least-Squares formulation of SND method in comparison to C-SVM and LS-SVM. All misclassification rates are collected on the identical test sets described in Section VII-A. Comparing the results in Tables II-VI we can clearly observe that our method is quite comparable in terms of generalization error to C-SVM and LS-SVM. In Tables V-VI we show p-values of a pairwise t-test which gives a clear evidence that generalization errors for SND and LS-SND are comparable to the corresponding values obtained for C-SVM and LS-SVM and there is no statistically significant difference in the mean values. However in Table III we can see that LS-SND algorithm almost in all cases is superior to LS-SVM and

[4]we report the percentage of the detected outliers

TABLE V
P-VALUES OF A PAIRWISE T-TEST ON GENERALIZATION ERROR BETWEEN SND AND OTHER METHODS

Dataset	to C-SVM	to LS-SVM
Toy Data (1)	0.87329	0.63883
Arcene	0.71842	0.52162
Ionosphere	0.73986	0.24175
Parkinsons	0.65938	0.97501
Sonar	0.47715	0.53844
Zoo	0.25673	0.011471
Iris	0.84167	0.84356
Ecoli	0.85788	0.02481
TAE	0.30483	1.9013e-09
Seeds	0.86329	0.20278

TABLE VI
P-VALUES OF A PAIRWISE T-TEST ON GENERALIZATION ERROR BETWEEN LS-SND AND OTHER METHODS

Dataset	to C-SVM	to LS-SVM
Toy Data (1)	0.8265	0.79085
Ionosphere	0.2189	0.00016358
Parkinsons	0.33084	0.40091
Sonar	0.8537	0.44872
Pima	0.82858	0.40384
Sonar	0.8537	0.44872
Zoo	0.0006965	0.90418
Iris	0.007038	0.068409
Ecoli	0.0039443	0.11129
TAE	0.75031	6.5273e-09
Seeds	0.18541	1

TABLE VII
P-VALUES OF A PAIRWISE T-TEST ON GENERALIZATION ERROR BETWEEN LARGE-SCALE PEGASOS-BASED SND AND OTHER METHODS

Dataset	to Pegasos	to NyFS-LSSVM
Pima	0.66776	9.5771e-22
Madelon	0.37543	1.4418e-08
Red Wine	0.45226	0.032591
White Wine	0.37445	9.4174e-20
Magic	9.3029e-08	1.0061e-07

obtains lower generalization errors. In general we can observe better performance from QP versions of SVM but this can be easily explained by properties of hinge-loss which better deals with the outliers. The latter disadvantage can be easily handled with a weighted formulation of LS-SVM [31].

TABLE VIII
EFFECTIVE VALUES OF THE l_2-NORMS AND THE $cos\theta$ VALUE BETWEEN THE CORRESPONDING NORMAL VECTORS IN FIGURE 5

Classes (c_i - c_j)	$\cos\theta$	norms ($\|w_i\|$, $\|w_j\|$)
c_1 - c_2	-0.3795	(0.5113, 0.4928)
c_1 - c_3	-0.4812	(0.5113, 0.5174)
c_2 - c_3	-0.4034	(0.4928, 0.5174)

For the second part of our numerical experiments we applied a large-scale modification of the SND algorithm to five large UCI datasets and collected corresponding misclassification errors. Table IV presents these results and we can see that almost everywhere NyFS-LSSVM [**?**] (Nyström Fixed-Size LS-SVM)

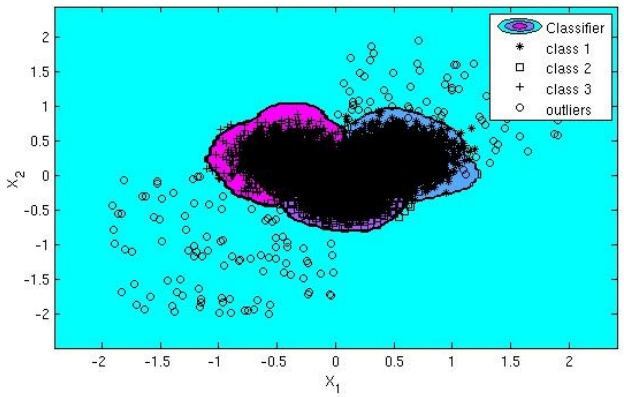


Fig. 5. Pegasos-based SND method in a novelty detection scheme with 3 classes. Size of the toy dataset is 9200.

TABLE IX
AVERAGED MISCLASSIFICATION ERROR / (DETECTION RATE) FOR SND AND ONE-CLASS SVM

Dataset	SND	One-Class SVM
Toy Data (2)	**0.0083** / (0.9746)	0.0233 / (1)
Toy Data (3)	**0.0113** / (1)	0.0233 / (1)
Toy Data (4)	**0.0366** / (0.8182)	0.0791 / (0.7808)

method achieves better performance than SND or Pegasos algorithms. This can be simply addressed by the nature of NyFS-LSSVM method, which is an exact algorithm while Algorithm 1 and Pegasos are approximate algorithms. On the other hand SND and Pegasos are very similar in the achieved results but for the largest Magic dataset SND surprisingly achieves better performance with very high statistical significance (see Table VII). One of the major advantages of Pegasos-based algorithms is the price of every iteration/training which can be controlled by m parameter in Algorithm 1. The example of novelty detection problem solved by this large-scale algorithm one can observe in Figure 5. Table VIII represents a pivot table of the effective values for the l_2-norms and the $cos\theta$ value between the corresponding normal vectors and decision boundaries (hyperplanes in the feature space) in Figure 5. This information helps us to understand the connection in a large-scale setting between the pairwise discrimination of classes and the corresponding compact support of the distributions from which these classes are drawn.

For the third part of our numerical experiments we have chosen to apply SND in an anomaly detection scheme in the presence of 2 or more classes. In this setting we cannot fairly compare our method to other SVM-based algorithms because of an obvious novelty of our problem. So we restrict ourselves to evaluating the SND algorithm for our 3 toy datasets and comparing it to One-Class SVM in terms of total misclassification error (assuming binary setting: non-outliers vs. outliers) and detection rate of outliers. From the Table IX we can clearly conclude that SND provides better support for underlying distributions and gives comparable or even better detection rates. One can also observe decision boundaries of the SND method for several random runs on different toy problems (Toy Data (2-4)) in Figure 6. The latter figure

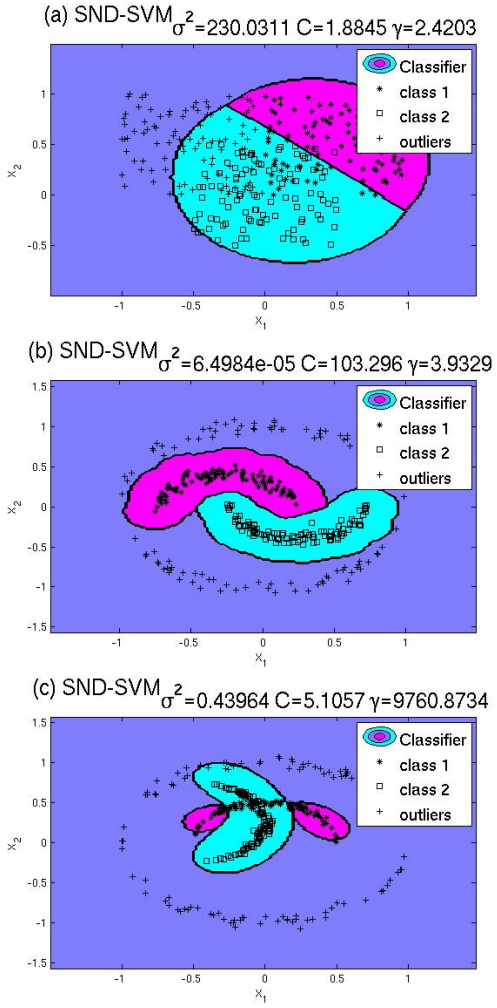


Fig. 6. SND method in a novelty detection scheme with 2 classes. Subfigures (a) through (c) represent SND boundaries in the presence of outliers (+) and correspond to Toy Data (2) through (4).

provides a better view on SND properties and output decision boundaries in the presence of the scattered outliers. In Figures 7 and 8 we can see a comparison of the SND approach with One-Class SVM. In Figure 7 we use for One-Class training all data points available in both classes while in Figure 8 we try to find the support for each class/distribution separately. Here by the white color we denote intersecting regions of two separate One-Class SVM estimators. However One-Class SVM is able to capture many data points by the underlying support it still far from the correct density estimation.

Analyzing these figures one can clearly observe the importance of labeling to capture the different underlying distributions in the data. One of the key advantages of the SND approach is a better understanding and modelling of the support for a mixture of distributions where one possesses a certain amount of information about each distribution.

C. Real life example

To justify the practical importance of our method we applied the SND Algorithm 1 in the context of AVIRIS data (Airborne Visible/InfraRed Imaging Sensor) [8]. We took one of the high definition greyscale images and extracted two disjoint sub-

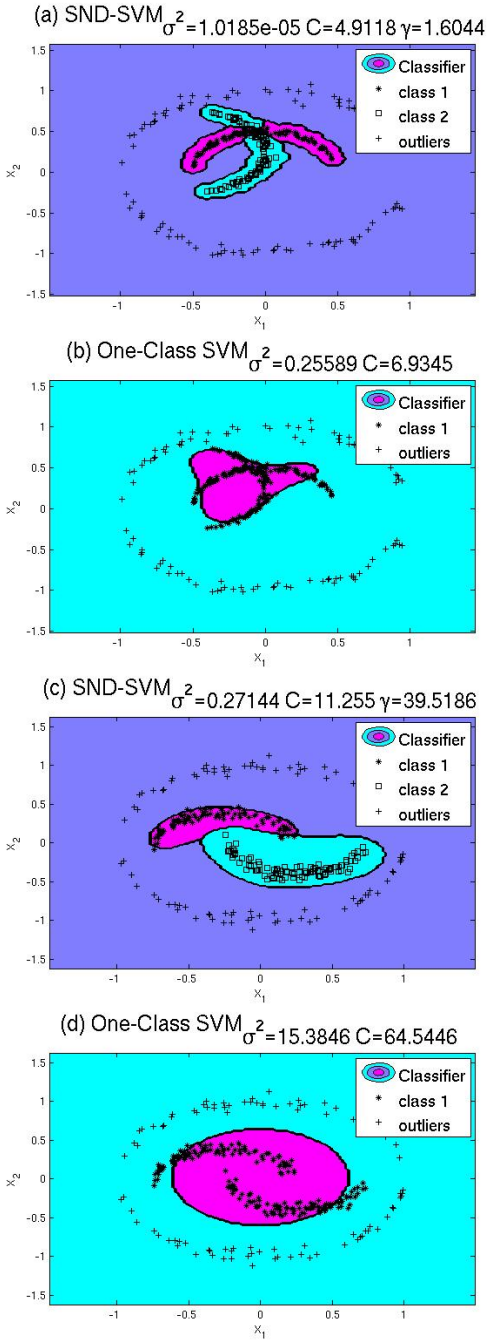


Fig. 7. Comparison of SND (a,c) and One-Class SVM (b,d) in the novelty detection scheme.

images of sizes 205×236 and 283×281 pixels respectively. The first sub-image was used for training the SND algorithm while the second one for test purposes.

We extracted for every pixel its intensity and averaged intensity of the window of size 10×10 of surrounding pixels excluding the nearest 5×5 pixels. Finally we took these values along with pixel intensities as our 2-dimensional training/test datasets. We separated the training image by the average white color intensity of the mentioned window across all pixels. Finally we defined outliers as the white spots on the darker

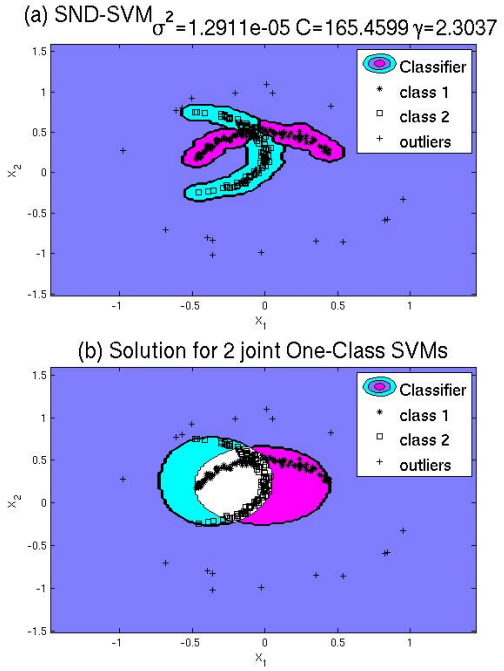


Fig. 8. Comparison of SND (a) and two joint One-Class SVMs (b) in the novelty detection scheme showing a clear improvement of SND. White region depicts the area which belongs to the support of both One-Class SVMs simultaneously.

greyscale region[5] by taking pixels belonging to that segment of the processed image with intensities grater than 190. The setting is artificial but it will help us to evaluate our approach *w.r.t.* real life data.

We applied Algorithm 3 to the final training data of size 48380 and determined σ parameter of the RBF kernel, λ and γ parameters of Algorithm 1 using 10-fold cross-validation on training data as described in Section VII-A. On every step of Algorithm 3 the SND model was calculated via Algorithm 1 and non-linearity of the model was achieved applying the Fixed-Size approach described in Section V-C.

In Figure 9 we can see these AVIRIS images while in Figure 10 we notice the same images but after the segregation to different terrains and detection of outliers by the SND and Pegasos[6] algorithms. As we can see our method is capable of good image segregation while being able to detect anomalous spots in the test image[7]. Both methods were able to detect outliers denoting pixels of interest[8] while Pegasos was much less accurate in estimating the densities of two classes and resulted in the increased number of the detected outliers[9]. These results can be extended to anomalous change detection when we consider the problem of finding anomalous changes in the obtained scenes of the same image.

In Figure 11 we can observe two histograms corresponding to the different decision functions obtained by SND Algorithm

[5]these spots correspond to the tracks remained after the transition of the fast boats

[6]we trained 2 Pegasos-based classifiers *w.r.t.* each class

[7]black pixels pointed by arrows in Figure 10

[8]big fast-boat transition track

[9]222 for SND and 507 for Pegasos

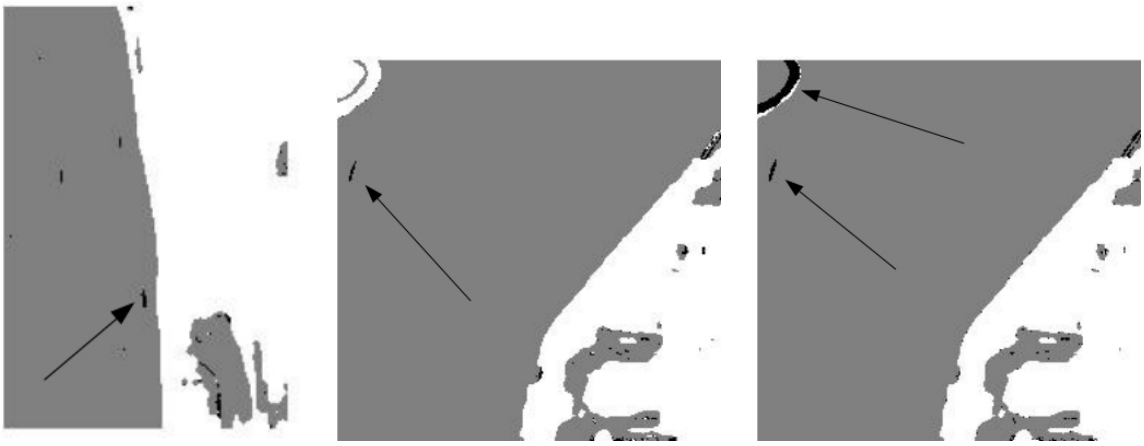


Fig. 10. AVIRIS training image after preprocessing (left) and test image after evaluation by the SND algorithm (middle) and the Pegasos algorithm (right) with pointed out outliers.

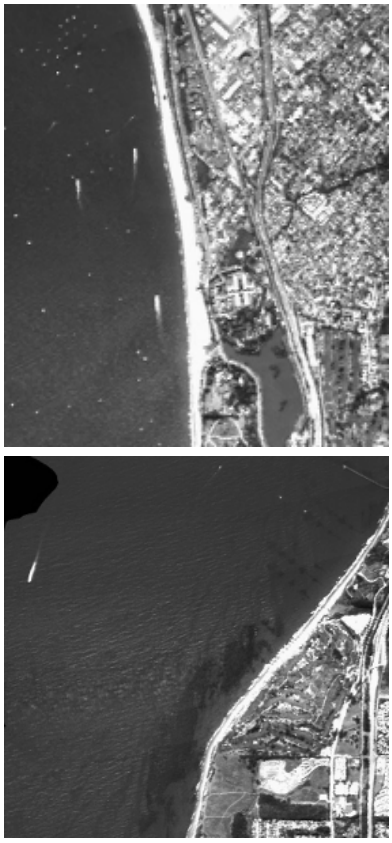


Fig. 9. AVIRIS training (top) and test (bottom) images.

3 which was evaluated on AVIRIS test image. Topmost image corresponds to the function which outputs positive values for the marine region and the bottom one outputs positive values for the land views. Analyzing these figures we can clearly notice some revealing patterns and distributions of output values. For instance in the images we can see two major peaks which obviously correspond to two classes. In general outliers are not concentrated as there are no intersecting peaks on both histograms. This fact corresponds to the intuition of [3] and validates the usefulness of the SND approach.

VIII. Conclusion and future work

In this paper we approached the novelty detection problem and estimation of the support for a high-dimensional distribution from the new perspective of multi-class classification. This setting is mainly designed for finding outliers in the presence of several classes while being valuable as a general purpose classifier as well. The SND setting can be potentially extended for a semi-supervised case with and an intrinsic norm [14] applied in conjunction with coupling terms (see Eq.(11)). The latter formulation implies that we need only few labeled data points to approximate the coupling term fairly well and the other data can be involved in the manifold learning. We consider the latter approach as a promising extension of our method for future work. We demonstrated that the performance and obtained generalization errors are comparable or even less than for other SVMs. The experimental results verify the usefulness of our approach for both settings: classification and novelty detection.

Acknowledgments

This work is supported by Research Council KUL, ERC AdG A-DATADRIVE-B, GOA/10/09MaNet, CoE EF/05/006, FWO G.0588.09, G.0377.12, SBO POM, IUAP P6/04 DYSCO. Johan Suykens is a professor at the KU Leuven, Belgium. The scientific responsibility is assumed by its authors. We wish to thank Gervasio Puertas for observations on the convexity of our dual objective in Eq.(18).

References

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

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[3] I. Steinwart, D. Hush, and C. Scovel, "A classification framework for anomaly detection," *J. Machine Learning Research*, vol. 6, pp. 211–232, 2005.

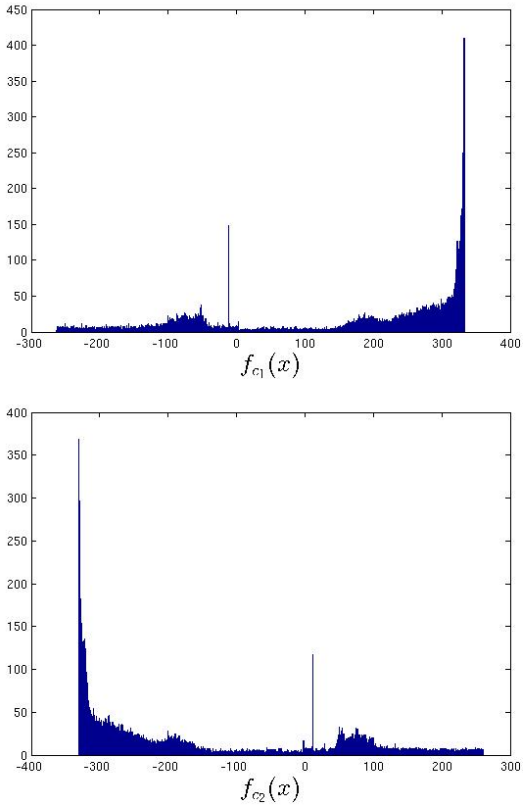


Fig. 11. Histograms of the output values for two decision functions (see Eq.(13)) obtained by SND Algorithm 3 and evaluated on AVIRIS test image. Top image corresponds to the function which outputs positive values for the marine region and the bottom one outputs positive values for the land views.

[4] C. Campbell and K. P. Bennett, "A Linear Programming Approach to Novelty Detection," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 395–401.

[5] M. J. Desforges, P. J. Jacob, and J. E. Cooper, "Applications of probability density estimation to the detection of abnormal conditions in engineering," in *Proceedings of Institute of Mechanical Engineers*, vol. 212, 1998, pp. 687–703.

[6] V. Jumutc and J. A. K. Suykens, "Supervised novelty detection," in *Proc. of the IEEE Symposium Series on Computational Intelligence*, 2013.

[7] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[8] N. Oppelt and W. Mauser, "The Airborne Visible / Infrared Imaging Spectrometer AVIS: Design, Characterization and Calibration," *Sensors*, vol. 7, no. 9, pp. 1934–1953, 2007. [Online]. Available: http://www.mdpi.com/1424-8220/7/9/1934/

[9] I. Steinwart, J. Theiler, and D. Llamocca, "Using support vector machines for anomalous change detection," in *IGARSS*, 2010, pp. 3732–3735.

[10] P. C. Hytla, R. C. Hardie, M. T. Eismann, and J. Meola, "Anomaly detection in hyperspectral imagery: comparison of methods using diurnal and seasonal data," *Journal of Applied Remote Sensing*, vol. 3, no. 1, pp. 033 546–033 546–30, 2009.

[11] L. Xu, K. Crammer, and D. Schuurmans, "Robust support vector machine training via convex outlier ablation," in *AAAI*, 2006, pp. 536–542.

[12] S. Melacci and M. Belkin, "Laplacian support vector machines trained in the primal," *Journal of Machine Learning Research*, vol. 12, pp. 1149–1184, 2011.

[13] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, "New support vector algorithms," *Neural Comput.*, vol. 12, no. 5, pp. 1207–1245, May 2000.

[14] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples." *Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[15] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, ser. COLT '92. New York, NY, USA: ACM, 1992, pp. 144–152.

[16] B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds., *Advances in kernel methods: support vector learning*. Cambridge, MA, USA: MIT Press, 1999.

[17] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *Proceedings of the 24th international conference on Machine learning*, ser. ICML '07, New York, NY, USA, 2007, pp. 807–814.

[18] J. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods – Support Vector Learning*, B. Schlkopf, C. Burges, and A. Smola, Eds. Cambridge, MA: MIT Press, 1998, pp. 42–65.

[19] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Math. Program.*, vol. 120, no. 1, pp. 221–259, 2009.

[20] K. De Brabanter, J. De Brabanter, J. A. K. Suykens, and B. De Moor, "Optimized fixed-size kernel models for large data sets," *Comput. Stat. Data Anal.*, vol. 54, no. 6, pp. 1484–1504, Jun. 2010.

[21] C. Williams and M. Seeger, "Using the nystrm method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*. MIT Press, 2001, pp. 682–688.

[22] T. Joachims, "Training linear svms in linear time," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '06. New York, NY, USA: ACM, 2006, pp. 217–226.

[23] S. Xavier-De-Souza, J. A. K. Suykens, J. Vandewalle, and D. Bollé, "Coupled simulated annealing," *IEEE Trans. Sys. Man Cyber. Part B*, vol. 40, no. 2, pp. 320–335, Apr. 2010.

[24] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, pp. 308–313, 1965.

[25] I. Steinwart, "On the influence of the kernel on the consistency of support vector machines." *Journal of Machine Learning Research*, vol. 2, pp. 67–93, 2001.

[26] A. Frank and A. Asuncion, "UCI machine learning repository," 2010. [Online]. Available: http://archive.ics.uci.edu/ml

[27] H. A. David and J. L. Gunnink, "The paired t test under artificial pairing," vol. 51, no. 1, pp. 9–12, Feb. 1997.

[28] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, May 2006.

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

[30] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[31] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: robustness and sparse approximation," *Neurocomputing*, vol. 48, no. 1, pp. 85–105, Oct. 2002.