Supervised Novelty Detection

Vilen Jumutc
K.U.Leuven, ESAT-SCD-SISTA
Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven (Heverlee), Belgium
Email: Vilen.Jumutc@esat.kuleuven.be

Johan A.K. Suykens
K.U.Leuven, ESAT-SCD-SISTA
Kasteelpark Arenberg 10, bus 2446
B-3001 Leuven (Heverlee), Belgium
Email: Johan.Suykens@esat.kuleuven.be

***Abstract*—In this paper we present a novel approach and a new machine learning problem, called Supervised Novelty Detection (SND). This problem extends the One-Class Support Vector Machine setting for binary classification while keeping the nice properties of novelty detection problem at hand. To tackle this we approach binary classification from a new perspective using two different estimators and a coupled regularization term. It involves optimization over a different objective and a doubled set of Lagrange multipliers. One might consider our approach as a joint estimation of the support for different probability distributions per class where an ultimate goal is to separate classes with the largest possible angle between the normal vectors to the decision hyperplanes in the feature space. Regarding an obvious novelty of our problem we report and compare the results along the lines of standard C-SVM, LS-SVM and One-Class SVM. Experiments have demonstrated promising results that validate the usefulness of the proposed method.**

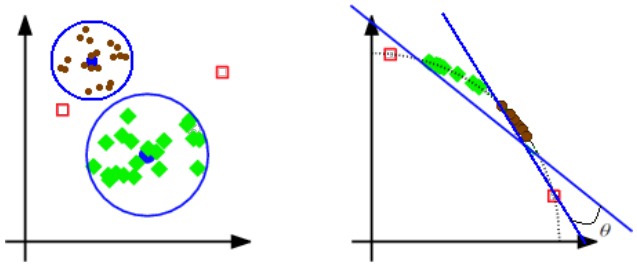

Fig. 1. SND solution in the input space (left) and feature space (right). SND aims at separating training data by maximizing the angle θ and separation of the classes. Maximization of the angle θ can be effectively substituted by the minimization of the inner product between normal vectors w_1 and w_2 to the decision hyperplanes.

I. Introduction

During many decades classification and novelty detection are among the most popular machine learning research areas. Separately these areas are studied quite well but together they never have been thoroughly considered. We close the gap in bringing together these machine learning paradigms.

Supervised Novelty Detection (SND) is designed for finding outliers[1] in the presence of several classes. For the proof of concept in this paper we mainly consider only the binary setting. SND can be effectively used for multi-class classification as well and it supplements the class of SVM-based algorithms. One can regard our approach as an extension of the original work by [1] for One-Class SVM where we deal with estimation of the support for a high-dimensional distribution. Contrary to Schölkopf's approach SND supports the *i.i.d.* assumption and density estimation for all target classes separately and it involves minimization over an additional coupling term between them. Altogether it brings a new kind of machine learning problem. One cannot view our problem as a natural extension of algorithms with "reject" option [2] because such methods deal with rejection in the sense of withdrawing from any answer regarding the evaluated sample while our approach clearly distinguishes between the considered classes and the outliers' class. On the other hand we may find some connections to [3] where the authors try to ablate outliers while trying to locate them via a reformulation of the SVM objective in terms of robust hinge loss. SND doesn't try to find outliers in the existing data pool. Our objective is to estimate the density of the two separate classes while trying to keep necessary discrimination between them.

[1] assuming out-of-sample evaluation

From a machine learning practitioner's perspective the SND algorithm could be utilized in many applications. Let us consider an example in the area of Intrusion Detection where several user groups are performing different actions on a system. Some of these users were identified as intruders or hackers afterwards. The fact that was confirmed by some statistical evidence. These users are effectively limited in number and cannot be considered as belonging to some specific class. Bringing this group as a separate class into a multi-class setting might not be very practical because of the initial diversity of its members and high risk of overfitting of the resulting classifier. Combining One-Class SVM with Multi-Class SVM might not be an optimal solution because of an added complexity and intermediate difficulties with integration in the provided solution. SND will provide a model for finding outliers (hackers) and affiliating a user with the most appropriate group. This application can be a part of Intrusion Detection System (IDS) but relies on the presence of several observed classes in contrast with the problem presented by [4], [5]. Another interesting application of SND can be found in bioinformatics where we have to locate misbehaving genes out of several possible groups in the genome while keeping necessary discrimination along these groups.

The remainder of this paper is structured as follows. Section II gives an overview of our method and the resulting optimization problem. Section III provides the experimental setup and results, while Section IV discusses some important issues and connections with respect to other approaches. Section V concludes the paper.

II. METHOD

A. Optimization problem

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

$$(x_1, y_1), \ldots, (x_l, y_l), x_i \in \mathcal{X}, y_i \in \{-1, 1\},$$

where l is the number of corresponding observations in the set $\mathcal{X}$. For simplicity we think of all our data as a compact subset of $\mathbb{R}^d Comeon1985@$. Then let Φ be a feature map $\mathcal{X} \to \mathcal{F}$ in connection to a positive definite Gaussian kernel [8], [9]

$$k(x, y) = \langle \Phi(x), \Phi(y) \rangle = e^{-\frac{\|x-y\|^2}{2\sigma^2}}. \tag{1}$$

Index i spans the range $\overline{1, l}$ if it is not declared explicitly. Greek letters $\alpha, \beta, \lambda, \xi$ without indices denote l-dimensional vectors.

In the remainder of this section, we develop an algorithm which returns two decision functions f_{c_1} and f_{c_2} for each of the involved target classes for a good classification. These functions should output positive values in a corresponding region capturing most of the data points belonging to their classes. According to classical work by [1] our aim is to map the data points into the feature space and to separate them from the origin with maximum margin. In addition to L2 regularization our assumption is based on a simple geometrical interpretation: to maximize the separation between the classes in the input space we minimize the cosine between the normal vectors of the corresponding hyperplanes in the feature space. While putting the explicit regularization on the cosine

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

is not practical we concentrate on minimizing only the numerator of the cosine. This setting leads to regularization on the inner product between w_1 and w_2.

Remark 1: Here we should emphasize that putting cosine in the optimization objective together with the norms of w_1 and w_2 will lead to a contradictory objective of minimizing $\langle w_1, w_2 \rangle$ and maximizing $\|w_1\| \|w_2\|$ where the final objective is not convex. On the other hand we need to trade-off two notions of separation and to find the most compact support of two classes while being able to discriminate them. We cope with the latter problem by introducing the additional parameter γ in Eq.(3). We will focus on a convex formulation of the problem which helps to tackle this trade-off.

First we start with the initial set of constraints to clarify the nature of our optimization problem w.r.t. normal vectors w_1, w_2 and maximization of the ρ bias term [1], [10]

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

To make a link between the One-Class SVM formulation and our method we combine the constraints in Eq.(2) and propose the following optimization problem

$$\begin{aligned} \min_{w_1, w_2 \in \mathcal{F}; \xi, \xi^* \in \mathbb{R}^l; \rho_1, \rho_2 \in \mathbb{R}} \quad & \tfrac{\gamma}{2}(\|w_1\|^2 + \|w_2\|^2) + \langle w_1, w_2 \rangle \\ & + C \textstyle\sum_{i=1}^l (\xi_i + \xi_i^*) - \rho_1 - \rho_2 \end{aligned} \tag{3}$$

$$\begin{array}{lll} \text{s.t.} & y_i(\langle w_1, \Phi(x_i) \rangle - \rho_1) + \xi_i \geq 0, & i \in \overline{1, l} \\ & y_i(\langle w_2, \Phi(x_i) \rangle - \rho_2) - \xi_i^* \leq 0, & i \in \overline{1, l} \\ & \xi_i \geq 0, \xi_i^* \geq 0, & i \in \overline{1, l} \end{array} \tag{4}$$

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

$$f_{c_1}(x) = \langle w_1, \Phi(x) \rangle - \rho_1, \tag{5}$$

$$f_{c_2}(x) = \langle w_2, \Phi(x) \rangle - \rho_2. \tag{6}$$

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

$$c(x) = \begin{cases} \text{argmax}_{c_i} \ f_{c_i}(x), & \text{if } \max_i f_{c_i}(x) > 0 \\ c_{out}, & \text{otherwise,} \end{cases} \tag{7}$$

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

Remark 2: 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 formulation in Eq.(3–4) 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 doesn't enter the optimization objective. This is different from Laplacian SVMs [6] and manifold regularization [7]. The data $\mathcal{Z}$ are a subset of $\mathcal{X}$ defined as follows

$$z_1, \ldots, z_m \in \mathcal{Z} \subseteq \{\mathcal{X} : c(x) = c_{out}\}, \tag{8}$$

where $c(x)$ is given by Eq.(7) and corresponds to the outliers' class.

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

$$\begin{aligned} \mathcal{L}(w_1, w_2, \xi, \xi^*, \rho_1, \rho_2, \alpha, \lambda, \beta, \beta^*) = & \tfrac{\gamma}{2}(\|w_1\|^2 + \|w_2\|^2) \\ & + \langle w_1, w_2 \rangle + C \textstyle\sum_{i=1}^l (\xi_i + \xi_j^*) \\ & - \textstyle\sum_{i=1}^l \alpha_i (y_i(\langle w_1, \Phi(x_i) \rangle - \rho_1) + \xi_i) \\ & + \textstyle\sum_{i=1}^l \lambda_i (y_i(\langle w_2, \Phi(x_i) \rangle - \rho_2) - \xi_i^*) \\ & - \textstyle\sum_{i=1}^l \beta_i \xi_i - \sum_{i=1}^l \beta_i^* \xi_i^* - \rho_1 - \rho_2. \end{aligned} \tag{9}$$

By setting the derivatives of Lagrangian with respect to the primal variables to zero we obtain

$$w_1 = \frac{\gamma \sum_{i=1}^l \alpha_i y_i \Phi(x_i) + \sum_{i=1}^l \lambda_i y_i \Phi(x_i)}{\gamma^2 - 1}, \tag{10}$$

$$w_2 = \frac{\gamma \sum_{i=1}^l \lambda_i y_i \Phi(x_i) + \sum_{i=1}^l \alpha_i y_i \Phi(x_i)}{1 - \gamma^2}, \tag{11}$$

$$\begin{array}{c} C - \beta_i - \alpha_i = 0, \quad \forall i \in \overline{1, l} \\ C - \beta_i^* - \lambda_i = 0, \quad \forall j \in \overline{1, l} \end{array} \tag{12}$$

$$\textstyle\sum_{i=1}^l \alpha_i y_i = 1, \quad \sum_{i=1}^l \lambda_i y_i = -1. \tag{13}$$

Substituting Eq.(10-13) into the Lagrangian and using the kernel trick with the expression given by Eq.(1) one can directly obtain the matrix form of the corresponding Lagrangian to be maximized

$$\max_{\alpha,\lambda} \mathcal{L}_D(\alpha,\lambda) = \frac{\mu_1}{2}(\alpha^T G\alpha + \lambda^T G\lambda) + \mu_2(\alpha^T G\lambda), \quad (14)$$

$$\begin{aligned} \text{s.t.} \quad & C \geq \alpha_i \geq 0, \ \forall i \\ & C \geq \lambda_i \geq 0, \ \forall i \\ & y^T\alpha = 1, \\ & y^T\lambda = -1, \end{aligned} \quad (15)$$

where y is a vector of labels, K is the kernel matrix of dimension $l \times l$ 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 componentwise multiplication. $\mathcal{L}_D$ is maximized and supplements the class of QP problems with box constraints. The expression for f_1 and f_2 becomes

$$f_{c_1}(x) = \frac{\gamma \sum_{i=1}^{l} \alpha_i y_i k(x_i, x) + \sum_{i=1}^{l} \lambda_i y_i k(x_i, x)}{\gamma^2 - 1} - \rho_1, \quad (16)$$

$$f_{c_2}(x) = \frac{\gamma \sum_{i=1}^{l} \lambda_i y_i k(x_i, x) + \sum_{i=1}^{l} \alpha_i y_i k(x_i, x)}{1 - \gamma^2} - \rho_2. \quad (17)$$

We can ensure the concavity of our dual objective in Eq.(14) 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.

B. Algorithm and further remarks

A qualitative example of the SND algorithm is given in Figure 1 where on the left subfigure one can see two minimal enclosing spheres for the two classes while with the □ we depict outliers. The left subfigure shows the resulting solution in the input space. On the other hand the right subfigure leads us to the feature space where we have to separate data points from the origin while minimizing the cosine of the angle θ between normal vectors w_1 and w_2 to the separating hyperplanes[2]. The latter objective effectively enables separation between the target classes.

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 1–2. One should notice that the main difference between both algorithms is the cross-validation step, decision rule and the input data.

Algorithm 1: SND for binary classification

input : training data X of size $l \times d$, class labels Y of size $l \times 1$
output: SND explicit decision rule

begin
  $[\gamma, \sigma, C] \leftarrow$ `CrossvalidateSND` (X, Y);
  $[\alpha, \lambda, \rho_1, \rho_2] \leftarrow$ `ComputeSND` $(X, Y, \gamma, \sigma, C)$;
  $c(x) \leftarrow \text{argmax}_{c_i} \ f_{c_i}(x)$;
end

[2]using simple geometry we can ensure that the angles between the hyperplanes and between the corresponding normal vectors are identical

Algorithm 2: 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 $l \times 1$
output: SND explicit decision rule

begin
  $[\gamma, \sigma, C] \leftarrow$ `CrossvalidateSND` (X, Y, Z);
  $[\alpha, \lambda, \rho_1, \rho_2] \leftarrow$ `ComputeSND` $(X, Y, \gamma, \sigma, C)$;
  $c(x) \leftarrow \begin{cases} \text{argmax}_{c_i} \ f_{c_i}(x), & \text{if } \max_i f_{c_i}(x) > 0 \\ c_{out}, & \text{otherwise} \end{cases}$;
end

In the above algorithms the "CrossvalidateSND" function stands for the tuning procedure which will be described in the next section. The crucial difference between Algorithm 1 and 2 is the usage of the data Z defined in Eq.(8). The SND model is tuned to perform novelty detection with respect to data Z and maximize the observed detection rate. Here we refer to Z as a matrix containing subset $\mathcal{Z} \subseteq \mathcal{X}$ which can be either labeled or not labeled. In case it is labeled, the labeling information can be used in the cross-validation procedure. 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 and λ, the primal variables ρ_1 and ρ_2, the optimal parameters γ, σ and the labeling Y in Eq.(16–17). Here we can notice that for Algorithm 1 we are not giving any alternative decisions in $c(x)$ and are obliged to select either class c_1 or c_2.

III. Experiments

A. Experimental setup

In all our experiments for all tested SVM-based models we use a 2-step procedure for tuning the parameters. The further described procedure for all experiments involves only training set. This procedure consists of Coupled Simulated Annealing [11] initialized with 5 random sets of parameters for the first step and the simplex method [12] 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 results tend to vary more because of the randomness in initialization.

We selected a universal RBF kernel (see [13]) that is generally capable to separate all compact subsets and is suitable for many kinds of data. The choice of RBF kernel was motivated by [1] where the authors claim an obvious advantage of it and that the data are always separable from the origin in feature space (see Definition 1 in [1]). We tune the bandwidth of RBF kernel in Eq.(1) with additional trade-off parameters for all methods using the tuning procedure described within the previous paragraph.

For the Toy Data (1) we performed 100 iterations with random sampling[3] according to separate uniform distributions from intersecting intervals $[0, 1]$ and $[-0.5, 0.5]$, collected sparsity and averaged error rates with corresponding ROC

[3]of size 100

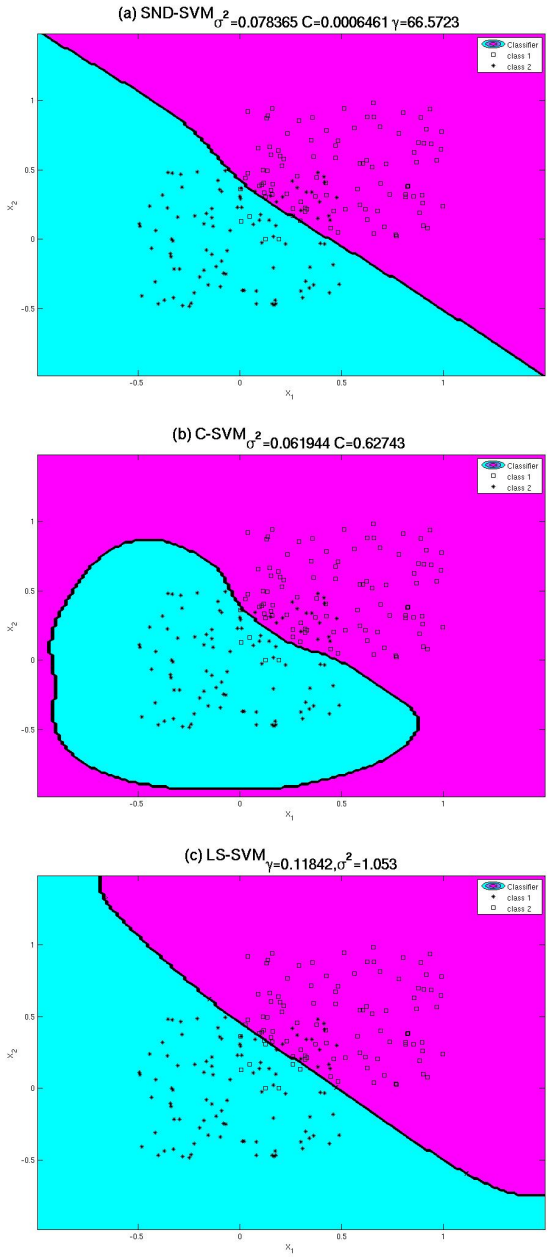

Fig. 2. Decision boundaries for binary classification problem: (a) SND, (b) C-SVM, (c) LS-SVM

curves. For novelty detection we performed 100 iterations with random sampling from three different distributions[4] (see Figure 4) scaled to the range $[-1, 1]$ for all dimensions. For all toy datasets in every iteration we splitted 100 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 [14] except for Arcene we used 10-fold splitting and performed averaging and paired t-tests [15] for the comparison of errors and achieved sparsity. Arcene was split to training and validation datasets initially and we simply run the classification scheme 10 times. For the properties of UCI and toy datasets used in this paper one can refer to the Table I.

[4]Toy Data (2-4)

TABLE I. DATASETS

Dataset	# of attributes	# of classes	# of data points
Toy Data (1)	2	2	200
Toy Data (2-4)	2	2	140
Arcene	10000	2	900
Ionosphere	34	2	351
Parkinsons	23	2	197
Sonar	60	2	208

We implemented the SND method as an optimization problem using the Ipopt package (see [16]), which implements a general purpose interior point search algorithm. For learning C-SVM, ν-SVM and One-Class SVM we used the LIBSVM package [17]. All the experiments were run on Core i7 CPU with 8GB of RAM available.

TABLE II. AVERAGED MISCLASSIFICATION ERROR

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

TABLE III. AVERAGED PERCENTAGE OF NON-ZERO α'S (SPARSITY)

Dataset	SND	C-SVM	LS-SVM
Toy Data (1)	0.5117± 0.106	**0.4599**± 0.158	1± 0
Arcene	**0.5220**± 0.070	0.9290± 0.027	1± 0
Ionosphere	**0.2287**± 0.048	0.3773± 0.110	1± 0
Parkinsons	0.5114± 0.036	**0.4894**± 0.094	1± 0
Sonar	**0.4730**± 0.025	0.6357± 0.082	1± 0

B. Numerical results

First we present some results for the binary classification setting where we can fairly compare our method to C-SVM [8] and LS-SVM [18]. Then we proceed with the novelty detection scheme in the presence of 2 classes and some number of outliers. Here we simply present preliminary results for different toy problems and report performance in terms of general test error and detection rate[5].

TABLE IV. 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

Comparing the results in Tables II-III we can clearly observe that our method is quite compatible in terms of generalization error to C-SVM and LS-SVM and produces sparser[6] solutions within the binary classification setting. The latter observation was quite surprising. It still needs to be investigated why L2-norm regularization with an additional

[5]we report the percentage of detected outliers

[6]in terms of non-zero dual variables

coupling term outperforms C-SVM in sparsity. For a better understanding we performed a joint experiment on Toy Data for ν-SVM [1] and found that our method is quite comparable in performance with it. Comparing ν-SVM average misclassification rate of 0.15 ± 0.086 and achieved sparsity of 0.653 ± 0.194 to values in Tables II-III (see Toy Data row) we can conclude that although the SND method is very close in sparsity to C-SVM it produces much sparser solutions than ν-SVM. In Figure 3 we can observe that the AUC values for different methods applied in our toy problem are quite comparable as well.

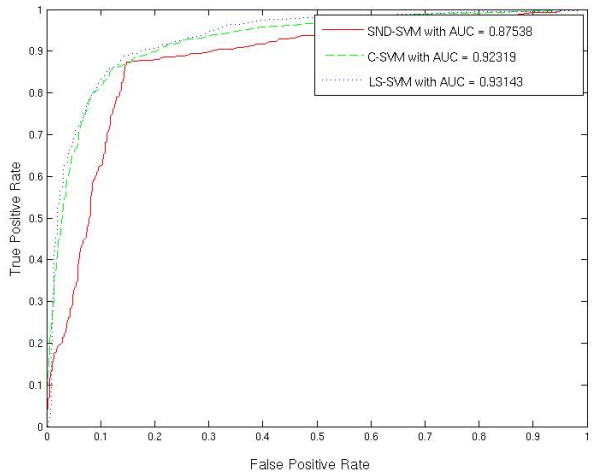

Fig. 3. ROC curves with AUC for Toy Data (1) problem.

In Table IV we show p-values of a pairwise t-test which gives a clear evidence that generalization error for SND is compatible to the corresponding value obtained for C-SVM and LS-SVM and there is no statistically significant difference in the mean value.

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

Dataset	SND	One-Class SVM
Toy Data (2)	**0.0239** / (0.9486)	0.0236 / (0.9965)
Toy Data (3)	**0.0432** / (0.9951)	0.1532 / (0.9342)
Toy Data (4)	**0.0700** / (0.8270)	0.1871 / (0.7090)

For the second part of our numerical experiments we have chosen to apply SND in an anomaly detection scheme in the presence of 2 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 V we can clearly conclude that SND provides better support for underlying distributions and gives comparable or even better detection rates. One can observe decision boundaries of the SND method for several random runs on different toy problems[7] in Figure 4. The latter figure provides a better view on SND properties and output decision boundaries in the presence of the scattered outliers.

[7]all the tuning parameters are estimated by the procedure described in Section III-A

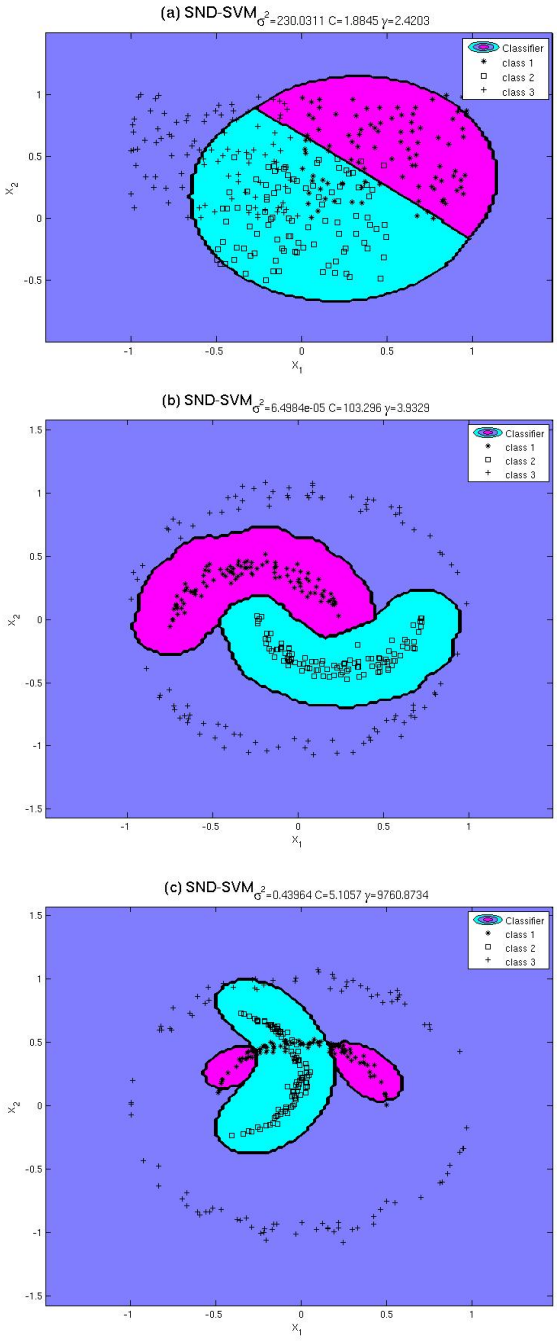

Fig. 4. SND method in a novelty detection scheme. Subfigures (a) through (c) represent SND boundaries in the presence of outliers (+).

IV. DISCUSSION

A. Difference with other SVMs

Hereby we can think of SND as solving a density estimation problem for each involved distribution per class while trying to separate the underlying classes as much as possible. In practice this results in finding an appropriate trade-off

between the amount of errors, separation and compactness[8] of our model describing these particular distributions. The demonstrated problem is not of the same kind as other SVMs where we cope only with optimal separation w.r.t. to regularization and amount of errors. For instance, in Laplacian SVMs [6] one uses additional regularization to keep the values of the decision function for adjacent points similar but this regularization mostly affects unlabeled samples.

B. Comparison with One-Class SVM

For the completeness of our research we want to present a comparison of the SND method to One-Class SVM on the identical toy problems in Figure 5. The data consist of ellipsoidal distributions with some random noise similar to Figure 4 (b-c). 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 we possess a certain amount of information about the distribution from which the data points are drawn. The SND setting can be effectively extended for a semi-supervised case with and an intrinsic norm [7] applied in conjunction with coupling terms (see Eq.(3)). 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.

Comparing SND and One-Class SVMs we can return to Figure 1 and see that the decision functions in the input space or the hyperplanes in feature space are strongly interfering while trying to accomplish 2-way separation. Firstly the hyperplanes separate the data points from the origin which is the connection to One-Class SVM. Secondly the hyperplanes try to maximize the angle between each other and locate data points in the separate halfspaces. The latter feature clearly distinguishes the SND approach from One-Class SVM.

C. Future work

Extending the current optimization problem for the case of n_c classes is considered to be a promising but challenging objective because it effectively involves optimization over $n \times n_c$ dual variables. This amount of variables significantly slows down every iteration of the Ipopt solver and starting from 500 data points even our approach for tuning the parameters (see Section III-A) becomes unfeasible. To tackle this problem one may study a scalable SMO-like method by Platt [19] or Nesterov's approach for convex optimization [20].

V. Conclusion

In this paper we approached the novelty detection problem and estimation of the support for a high-dimensional distribution from the new perspective of binary 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. We demonstrated that the obtained sparsity and 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.

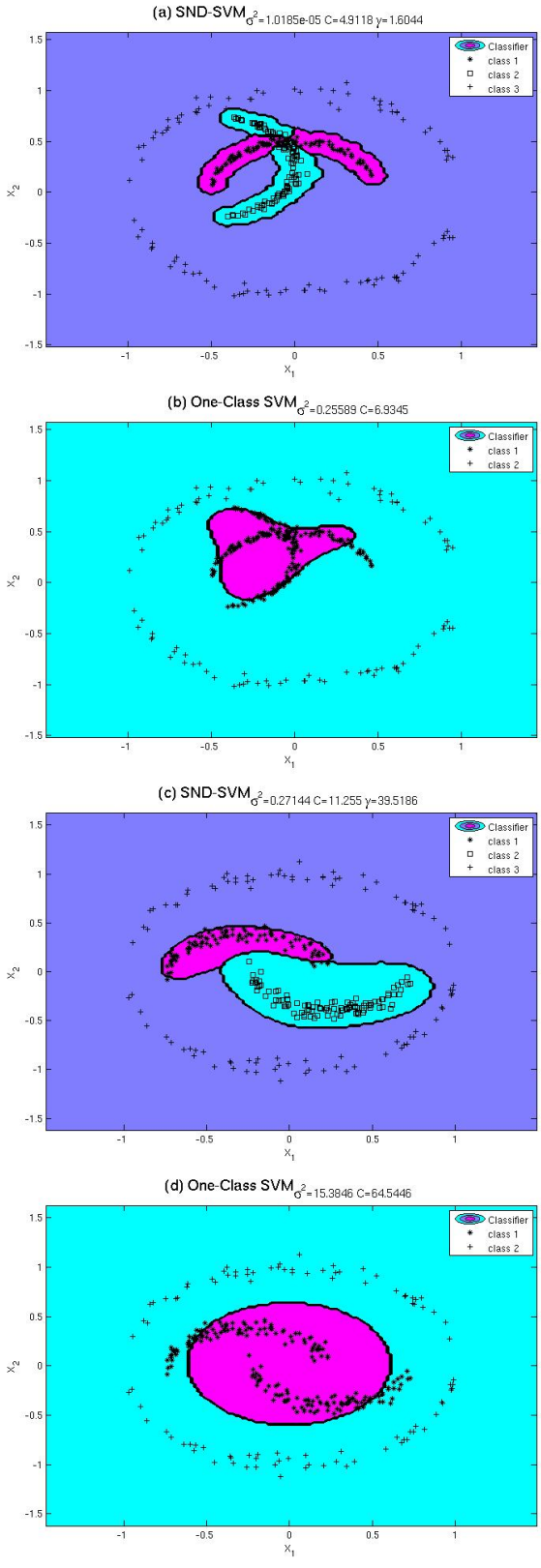

Fig. 5. Comparison of SND (a,c) and One-Class SVM (b, d) in the novelty detection scheme.

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

ACKNOWLEDGMENTS

This work was supported by grants and projects for the Research Council K.U.Leuven (GOA-Mefisto 666, GOA-Ambiorics, several PhD / Postdocs & fellow grants), ERC Advanced Grant A-DATADRIVE-B, the Flemish Government FWO: PhD / Postdocs grants, projects G.0240.99, G.0211.05, G.0407.02, G.0197.02, G.0080.01, G.0141.03, G.0491.03, G.0120.03, G.0452.04, G.0499.04, G.0226.06, G.0302.07, ICCoS, ANMMM; AWI;IWT:PhD grants, GBOU (McKnow) Soft4s, the Belgian Federal Government (Belgian Federal Science Policy Office: IUAP V-22; PODO-II (CP/01/40), the EU(FP5-Quprodis, ERNSI, Eureka 2063-Impact;Eureka 2419-FLiTE) and Contracts Research/Agreements(ISMC/IPCOS, Data4s, TML, Elia, LMS, IPCOS, Mastercard). Johan Suykens is a professor at the K.U.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.(14).

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] C. Chow, "On optimum recognition error and reject tradeoff," *IEEE Trans. Inf. Theor.*, vol. 16, no. 1, pp. 41–46, Sep. 2006.

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

[4] K. A. Heller, K. M. Svore, A. D. Keromytis, and S. J. Stolfo, "One class support vector machines for detecting anomalous windows registry accesses," in *Proc. of the workshop on Data Mining for Computer Security*, 2003.

[5] S. J. Stolfo, F. Apap, E. Eskin, K. Heller, S. Hershkop, A. Honig, and K. Svore, "A comparative evaluation of two algorithms for windows registry anomaly detection," *J. Comput. Secur.*, vol. 13, no. 4, pp. 659–693, Jul. 2005.

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

[7] 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.

[8] 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.

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

[10] 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.

[11] 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.

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

[13] 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.

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

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

[16] 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.

[17] 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.

[18] 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.

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

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