One-class Classification: ν-SVM

Qiang Ning

Dec. 10, 2015

Abstract

One-class classification is a special kind of classification problem, for which the training set only consists of samples from one class. Conventional SVM fails to handle the one-class classification problem because of the lack of information from the other class. The ν-SVM addresses this issue by estimating the probability density support of the class that we have sufficient samples, and then treat new samples outside of the support as outliers. The resulting optimization problem can be readily solved in a similar way as the conventional SVM, and its generalization error can also be theoretically upper bounded. Both simulation and real medical data are used to demonstrate the performance of ν-SVM in this report, which should prove useful in various outlier/abnormity detection tasks.

1 Introduction

Classification is to differentiate objects and understand information. When the underlying probability distribution is readily available, classification tasks can be easily handled within the Bayesian framework. For instance, in binary classification/detection, given prior distribution π_y, $y = \{\pm 1\}$, and conditional distribution $p_y(\mathbf{x})$, $y = \{\pm 1\}$, where $\mathbf{x} \in \mathbb{R}^d$ is observation and y is class label, the optimal classifier that minimizes the "0-1" loss is a likelihood ratio test:

$$\delta_B(\mathbf{x}) = \begin{cases} 1 & \text{if } L(\mathbf{x}) \geq \eta \\ -1 & \text{otherwise} \end{cases},$$

where $L(\mathbf{x}) = p_1(\mathbf{x})/p_{-1}(\mathbf{x})$ is the likelihood ratio function, and $\eta = \pi_{-1}/\pi_1$ is the testing threshold [1].

In practice, however, the underlying probability distribution is usually unavailable due to the lack of knowledge about the physical and statistical law governing different classes of observations. On the other hand, observation data can often be easily collected. Therefore, it has been proposed to "learn" a classifier based on existing observations (i.e., the training dataset), with the hope/assumption that a classifier that separates the training dataset well can also classify future observations (i.e., the test dataset) well. Various classification methods have been proposed along

this way: empirical risk minimization (ERM), support vector machine (SVM), logistic regression, neural network, etc. [2]

Nevertheless, in some real world applications, e.g., outlier detection, not only the underlying probability distribution is unavailable, but it is also very expensive or even impossible to collect data from both two classes. As a result, the training set only consists of data from one class (or the data from the other class are insufficient). The classification problem in this scenario is often called the one-class classification problem. The so-called ν-SVM, which we are going to explore in this report, is one of the popular methods to solving this problem [3]. Throughout this report, we would refer binary and multi-class classification problem as the conventional classification problem.

2 Challenges

As implied by its name, one-class classification problem is challenging, because no (or insufficient) information about the outliers is available and conventional classification methods cannot be used.

To better illustrate this point, we take the conventional SVM (here we focus on the maximum-margin classifier) as an example. As in [2], the maximum-margin classifier is to construct a classifier $\delta : \mathbb{R}^d \mapsto \{\pm 1\}$ such that

$$\delta(\mathbf{x}) = \operatorname{sgn}\left[g(\mathbf{x})\right],$$

where the discriminant function $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x}+w_0$.[1] Given a training set with n samples, $\{\mathbf{X}_i, Y_i\}_{i=1}^n$, where $\mathbf{X}_i \in \mathbb{R}^d$ and $Y_i \in \{\pm 1\}$, for all i, the weight vector $\mathbf{w}$ and bias w_0 are obtained through solving the following optimization problem:

$$\min_{\mathbf{w},w_0} \frac{1}{2}\|\mathbf{w}\|_2^2$$
$$\text{s.t. } y_i\left[\mathbf{w}^T\mathbf{x}_i + w_0\right] \geq 1,\ i = 1, \ldots, n.$$

If all the training samples are from class "+1", i.e., $y_i = 1$ for all i, then obviously the solution is $\mathbf{w}^\star = 0$ and any $w_0^\star \geq 1$, and the resulting classifier is $\delta(\mathbf{x}) \equiv 1$. Therefore, if we directly apply the conventional SVM to one-class classification, the resulting classifier will have no power in identifying outliers.

This failure of applying conventional SVM (and other conventional classification methods as well) to one-class classification can be explained by the fact that the conventional classification methods are designed to "separate" different classes. When no or few training samples are from class "-1", separation can be trivially satisfied, and the generalizability of the trained classifier is thus poor. Conceptually speaking, in conventional classification methods, description about one class is learnt via comparison to other classes, rather than the class itself. In one-class classification, the problem becomes challenging because we need to learn a description about a class itself. An

[1] We can also play the kernel trick here, i.e., replacing $\mathbf{x}$ by $\Phi(\mathbf{x})$, where $\Phi : \mathbb{R}^d \mapsto \mathbb{R}^k$ is a mapping from input space to feature space.

extreme point is to estimate the probability density from a training set, which would then allow us to solve whatever outlier detection problems. However, probability density estimation itself is still an open problem in the learning theory. One of the major drawbacks of probability density methods is the requirement for a large training set, especially when dealing with high-dimensional features.

To address this issue, ν-SVM is proposed in [3], which turns to solve an alternative problem: probability density support estimation. It learns a domain description about the one-class training set, and then use the domain description to detect outliers. The generalization error of ν-SVM can also be bounded theoretically.

3 One-class Classification: ν-SVMs

Following Vapnik's principle that never to solve a problem that is more general than the one we actually need to solve, ν-SVM is actually estimating the support of the probability density (i.e., a "smallest" region), instead of estimating the probability density. Specifically, the ν-SVM method is to separate the data from the origin with maximum margin (that is where the "SVM" in its name comes from). The strategy is to find a "smallest" region capturing most of the data points, so that within that region, the classifier decides "1", and otherwise decides "-1" (outlier). Next we describe the ν-SVM method by its formulation and algorithm.

3.1 Formulation

Given a training set with n samples, $\{\mathbf{x}_i\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$, ν-SVM is to solve the following problem:

$$\min_{\mathbf{w},\rho,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{1}{\nu n}\sum_{i=1}^{n}\xi_i - \rho \tag{1}$$
$$\text{s.t. } \mathbf{w}^T\Phi(\mathbf{x}_i) \geq \rho - \xi_i, \xi_i \geq 0,\ i = 1, \ldots, n,$$

where $\nu \in (0, 1]$, and $\Phi(\cdot)$ is the transformation from input space to feature space. The decision function is

$$\delta(\mathbf{x}) = \operatorname{sgn}\left[g(\mathbf{x})\right], \tag{2}$$

where the discriminant function $g(\mathbf{x}) = \mathbf{w}^T\Phi(\mathbf{x}) - \rho$. By formulating Eq. (1), we are expecting that for most training samples, the discriminant function is positive, while maintaining a small value of $\frac{1}{2}\|\mathbf{w}\|_2^2 - \rho$. The trade-off between these two goals is controlled by ν.

Let us first assume the slack variables ξ_i are zero, which would be true when $\nu = 0$. One significant difference between ν-SVM and SVM is the introduction of ρ. To understand why the introduction of ρ leads to a desirable classifier, we see that in Fig. 1, it can be derived that

$$d = \frac{|\rho|}{\|\mathbf{w}\|}.$$

The minimization of $-\rho$ is equivalent to the maximization of ρ. If a data point lies above the line (e.g., point A in Fig. 1), then $\rho > 0$, and a larger ρ indicates a larger d; if a data point lies below the line (e.g., point B in Fig. 1), then $\rho < 0$, and a larger ρ indicates a smaller d. In both cases, the discriminant line is moving toward the data point. Therefore, it can be seen that the introduction of ρ leads to a discriminant function that tightly bounds the training set.

Additionally, to handle the case where there are outliers in training set, slack variables ξ_i are introduced, similarly to what we did for soft-margin SVM [2]. As stated earlier, the trade-off between data consistency and boundary tightness is controlled by ν, but actually ν is more than simply a regularization parameter, which will be shown later in this report.

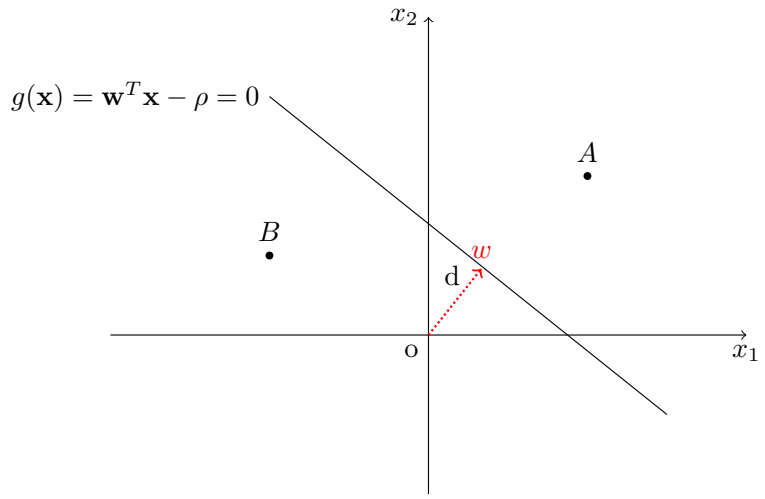

Figure 1: The normal vector of discriminant boundary $g(\mathbf{x}) = 0$ is $\mathbf{w}$. The distance d from origin to the boundary is thus $d = |\rho|/\|\mathbf{w}\|$. If point A lies in the region $g(\mathbf{x}) > 0$, then origin should satisfy $g(0) = -\rho < 0$, and ρ is thus positive; if point B lies in the region $g(\mathbf{x}) > 0$, then origin should satisfy $g(0) = -\rho > 0$, and ρ is thus negative.

3.2 Dual Problem

Problem Eq. (1) is referred to as the primal optimization problem. As what we did for conventional SVM, it is usually preferable to deal with its dual problem.

Firstly, we introduce a Lagrangian with $\alpha_i, \beta_i \geq 0$

$$L(\mathbf{w}, \boldsymbol{\xi}, \rho, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{w}\|_2^2 + \frac{1}{\nu n}\sum_i \xi_i - \rho - \sum_i \alpha_i(\mathbf{w}^T\Phi(\mathbf{x}_i) - \rho + \xi_i) - \sum_i \beta_i\xi_i, \tag{3}$$

whose first derivatives w.r.t. the primal variables $\mathbf{w}$, $\boldsymbol{\xi}$ and ρ are

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= \mathbf{w} - \sum_i \alpha_i \Phi(\mathbf{x}_i) \\
\frac{\partial L}{\partial \xi_i} &= \frac{1}{\nu n} - \alpha_i - \beta_i, i = 1, \ldots, n, \\
\frac{\partial L}{\partial \rho} &= -1 + \sum_i \alpha_i.
\end{aligned}$$

Then by setting these derivatives to zero, we have

$$\mathbf{w} = \sum_i \alpha_i \Phi(\mathbf{x}_i), \tag{4}$$

$$\alpha_i = \frac{1}{\nu n} - \beta_i \leq \frac{1}{\nu n}, i = 1, \ldots, n, \tag{5}$$

$$\sum_i \alpha_i = 1. \tag{6}$$

Substituting Eq. (4), Eq. (5) and Eq. (6) into Eq. (3), we obtain the dual problem:

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \\
&\text{s.t. } 0 \leq \alpha_i \leq \frac{1}{\nu n}, i = 1, \ldots, n, \text{ and } \sum_i \alpha_i = 1,
\end{aligned} \tag{7}$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ is the kernel function. As the primal problem Eq. (1), Eq. (7) is also a quadratic programming. Fast iterative algorithms exist for the dual problem Eq. (7). An algorithm originally proposed for classification is the so-called sequential minimal optimization (SMO) algorithm [4]. To solve Eq. (7) specifically, a modified version of SMO is proposed and can be found in [3][5].

Once an optimizing $\boldsymbol{\alpha}^\star$ is obtained by solving Eq. (7), we can recover $\mathbf{w}^\star$ using Eq. (4). As for ρ, we notice the fact that the constraints in Eq. (1) become equalities if α_i and β_i are positive, i.e., $0 < \alpha_i < \frac{1}{\nu n}$. Pick any one of such indices i, then

$$\rho^\star = (w^\star)^T \Phi(\mathbf{x}_i) = \sum_{j=1}^{n} \alpha_j^\star k(\mathbf{x}_j, \mathbf{x}_i).$$

4 Theory

Very nice theoretical results have been proven in [3]. In this report, we focus on two of the theorems introduced in [3], and go through the proofs. For Theorem 1, an alternative proof is provided instead of the original one provided in [3]. For Theorem 2, we fill up the blanks that the authors left behind and correct typos.

Theorem 1 (ν-Property)**.** *Assume the solution to Eq.* (1) *satisfies* $\rho \neq 0$*. The following statements hold:*

1. ν *is a lower bound on the fraction of support vectors.*

2. ν *is an upper bound on the fraction of outliers.*

Proof. To prove the two properties, the authors of [3] used a proposition which relates the $\mathbf{w}$ and ρ obtained in one-class classification with those obtained in corresponding binary classification.

Here, however, we can prove them alternatively as follows. Let $\mathcal{I} = \{i : \alpha_i \neq 0\}$. From Eq. (5) and (6), we have

$$1 = \sum_{i=1}^{n} \alpha_i = \sum_{i \in \mathcal{I}} \alpha_i = \frac{|\mathcal{I}|}{\nu n} - \sum_{i \in \mathcal{I}} \beta_i \leq \frac{|\mathcal{I}|}{\nu n}.$$

Therefore, $|\mathcal{I}| \geq \nu n$, i.e., the number of nonzero α_i's is lower bounded by νn. Note that nonzero α_i's correspond to support vectors, so property 1 holds.

Let $\mathcal{J} = \{j : \beta_j = 0\}$. Again from Eq. (5) and (6), we have

$$1 = \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \frac{1}{\nu n} - \beta_i = \frac{|\mathcal{J}|}{\nu n} + \sum_{j \notin \mathcal{J}} \alpha_j \geq \frac{|\mathcal{J}|}{\nu n}.$$

Therefore, $|\mathcal{J}| \leq \nu n$, i.e., the number of zero β_i's is upper bounded by νn. Note that zero β_i's correspond to outliers ($\xi_i > 0$), so property 2 holds.

□

Besides the ν-property which reveals the underlying meaning of regularization parameter ν, the learning generalizability of ν-SVM in terms of probability density support estimation can also be characterized as follows.

Definition 1. *Let* $f : \mathcal{X} \mapsto \mathbb{R}$*. For a fixed* $\theta \in \mathbb{R}$ *and* $\mathbf{x} \in \mathcal{X}$*, let* $d(\mathbf{x}, f, \theta) = \max\{\theta - f(\mathbf{x}), 0\}$*. Then for a training set* $T = \{\mathbf{x}_i\}_{i=1}^{n}$*, define*

$$\mathcal{D}(T, f, \theta) = \sum_{\mathbf{x} \in T} d(\mathbf{x}, f, \theta).$$

Theorem 2 (Generalization Error Bound)**.** *Assume we are given a training set* $T = \{\mathbf{x}_i\}_{i=1}^{n}$ *generated i.i.d. from an underlying but unknown distribution* P *which does not contain discrete components. Suppose a function* $f_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T \Phi(\mathbf{x})$ *and bias* ρ *are obtained by solving the optimization problem Eq.* (1)*. Let* $R_{\mathbf{w},\rho} = \{\mathbf{x} : f_{\mathbf{w}}(\mathbf{x}) \geq \rho\}$ *denote the decision region. Then with probability* $1 - \delta$ *over the draw of a random sample from* P*, for any* $\gamma > 0$*,*

$$P\{\mathbf{x}' : \mathbf{x}' \notin R_{\mathbf{w},\rho-\gamma}\} \leq \frac{2}{n}(k + \log_2 \frac{n^2}{2\delta}), \tag{8}$$

where

$$k = \frac{c_1 \log_2(c_2 \hat{\gamma} n)}{\hat{\gamma}^2} + \frac{2\mathcal{D}}{\hat{\gamma}} \log_2 \left(e \left(\frac{(2n-1)\hat{\gamma}}{2\mathcal{D}} + 1 \right) \right) + 2, \tag{9}$$

$c_1 = 16c^2$*,* $c_2 = \ln 2/(4c^2)$*,* $c = 103$*,* $\hat{\gamma} = \gamma/\|\mathbf{w}\|$*, and* $\mathcal{D} = \mathcal{D}(T, f_{\mathbf{w}}, \rho)$*.*

A training set T determines a decision region $R_{\mathbf{w},\rho}$, so that if a new sample falls into $R_{\mathbf{w},\rho}$, we assert it is generated from distribution P; otherwise, we assert it is an outlier. We make such assertions because we expect that points generated according to P will indeed lie in $R_{\mathbf{w},\rho}$. Theorem 2 gives us such a guarantee that with a certain probability (i.e., $1-\delta$), the probability of a new sample lies outside of the region $R_{\mathbf{w},\rho-\gamma}$ is bounded from above. Moreover, Theorem 2 also serves as a characterization of ν-SVM, from which we can gain the following insights.

1. The theorem suggests not to directly use the offset ρ obtained by solving Eq. (1), but a smaller value $\rho-\gamma$, which corresponds to a larger decision region $R_{\mathbf{w},\rho}$.

2. If $\mathcal{D}=0$, then as $n\to\infty$, the bound in Eq. (8) goes to zero, i.e., the complete support is obtained asymptotically. However, $\mathcal{D}$ is measured with respect to ρ, while the bound applied to a larger region $R_{\mathbf{w},\rho-\gamma}$. Any point in $R_{\mathbf{w},\rho-\gamma}-R_{\mathbf{w},\rho}$ will contribute to $\mathcal{D}$. Therefore, $\mathcal{D}$ is strictly positive, and this bound does not imply asymptotic convergence to the true support.

3. The existence of ν is to allow outliers in the training set, and to improve robustness. Since a larger ν indicates a larger $\mathcal{D}$, hence a larger k, we see that an unecessarily large ν will lead to a larger bound. Therefore, prior knowledge about the percentage of outliers in the training set is desired.

The proof of Theorem 2 requires concepts of covering number and function spaces, and can be found in the Appendix.

5 Experiments

A comprehensive off-the-shelf package for SVM is the LIBSVM [6], in which ν-SVM is also available.

5.1 ν-Property

In this section, we wish to verify the ν-property in Theorem 1. A crescent-shaped two-dimensional simulation dataset from [7] is used, of which the 500 samples are shown in Fig. 2.

An example of using ν-SVM to do one-class classification is shown in Fig. 3, where the Gaussian kernel

$$k(\mathbf{x},\mathbf{y})=e^{-0.06\|\mathbf{x}-\mathbf{y}\|^2}$$

was used, and ν was 0.05. We can see that a smooth, crescent-shaped decision boundary (blue curve) was learned, which tightly bounds a large portion of the training samples, while allowing a certain portion of outliers (black stars).

Using the same kernel function, the fraction of support vectors (SVs) and outliers (OLs) given different values of ν is summarized in Table 1 to verify Theorem 1. It can be seen from Table 1 that the fraction of SVs was lower bounded by ν. Moreover, the fraction of OLs is approximately

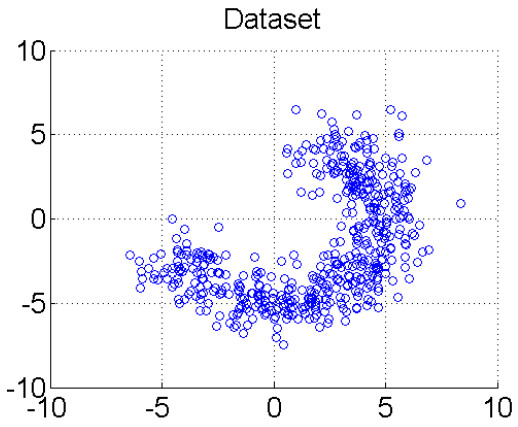

Figure 2: A simple 2-dimensional dataset with 500 samples from [7]. Blue circle: training sample.

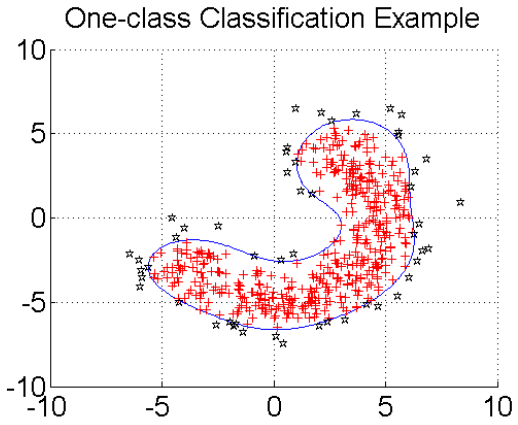

Figure 3: An example of using ν-SVM to learn a smallest region that captures most of the points. Blue curve: decision boundary obtained. Black star: outliers.

upper bounded by ν, inspite of some small fluctuations (e.g., when $\nu = 30\%, 70\%$) which can be explained by the fact that we are not in the asymptotic regime. Table 1 does indicate that ν can be used to approximate/control the fraction of SVs and OLs.

Table 1: The fraction of SVs and OLs given different values of ν.

ν (%)	Fraction of SVs (%)	Fraction of OLs (%)
5	6.2	5.0
10	11.0	10.0
30	31.6	30.2
50	50.2	49.8
70	70.2	70.2
90	90.2	90.0

5.2 Breast Cancer Classification

A dataset retrieved from the Wisconsin Breast Cancer Databases from UCI is used to demonstrate the performance of ν-SVM. [2] It contains 699 instances in total collected between 1989 and 1991 by Dr. WIlliam H. Wolberg at University of Wisconsin Hospitals [8], 458 instances of which correspond to benign breast cancer, and 241 of which malignant. The dimensionality of feature space is 9: clump thickness, uniformity of cell size, uniformity of cell shape, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses, all quantized from 1 to 10. Figure 4 is the scatter plot of the dataset after dimensionality reduction by PCA.

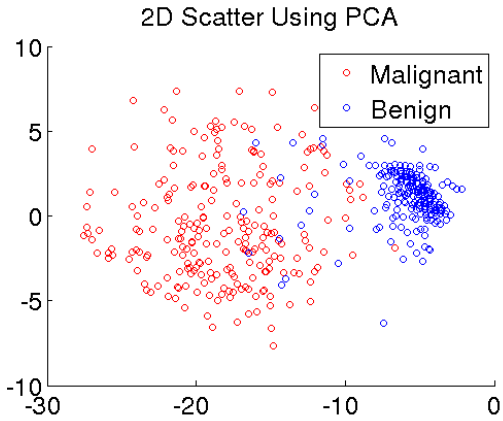

Figure 4: Scatter plot of the breast cancer dataset. Original data were projected onto the first two principal eigenbases of the empirical covariance marix. A natural clustering of benign and malignant can be observed.

[2] Link to data: `http://homepage.tudelft.nl/n9d04/occ/505/oc_505.mat`

Table 2 summarizes the performance of ν-SVM compared with conventional binary SVM. When the training set has an insufficient number of malignant instances, there are usually two options to do classification. One is to still train conventional SVMs using the whole dataset, and the other one is to alternatively train ν-SVMs using only the benign instances in the training set. By comparison between the first two rows, we can tell that it is better in terms of detecting malignant cancers if we only use benign instances and train ν-SVMs.

By comparing ν-SVM (row 2) with row 3-5 in Table 2, we can also see that when the size of the training set remains the same, one may prefer using one-class classification if the detection of outliers is more important, unless sufficient numbers of samples are available for both classes (e.g., row 6). Figure 5 provides a visual explanation to why ν-SVM is a better choice when dealing with "unbalanced" learning tasks. Therefore, we can see the importance of using one-class classification when information from one class is insufficient.

Table 2: The performance of ν-SVM. The left two columns represent the number of benign/malignant instances used in the training set. The right two columns are the probability of detection of benign cancer, and the probability of detection of malignant cancer, respectively. The row in bold face represents ν-SVM.

# Benign	# Malignant	Detection of Benign (%)	Detection of Malignant (%)
300	20	100.0	87.2
300	**0**	**97.5**	**96.5**
290	10	100.0	45.4
280	20	100.0	87.9
270	30	99.4	96.5
200	100	99.4	97.9

6 Discussion

We have seen the importance of using one-class classification methods to deal with the learning tasks where the training set only has one class. As one popular one-class classification method, ν-SVM can be proved to be equivalent to another method named SVDD: Support Vector Domain Description [9].

6.1 SVDD

Suppose the description about a data set $T = \{\mathbf{x}_i\}_{i=1}^n$ is required. While ν-SVM is to bound the data set using hyperplanes, SVDD is to use spheres instead. Specifically, we wish to find a "smallest" ball that most of the data points in T can be put into. The resulting primal optimization

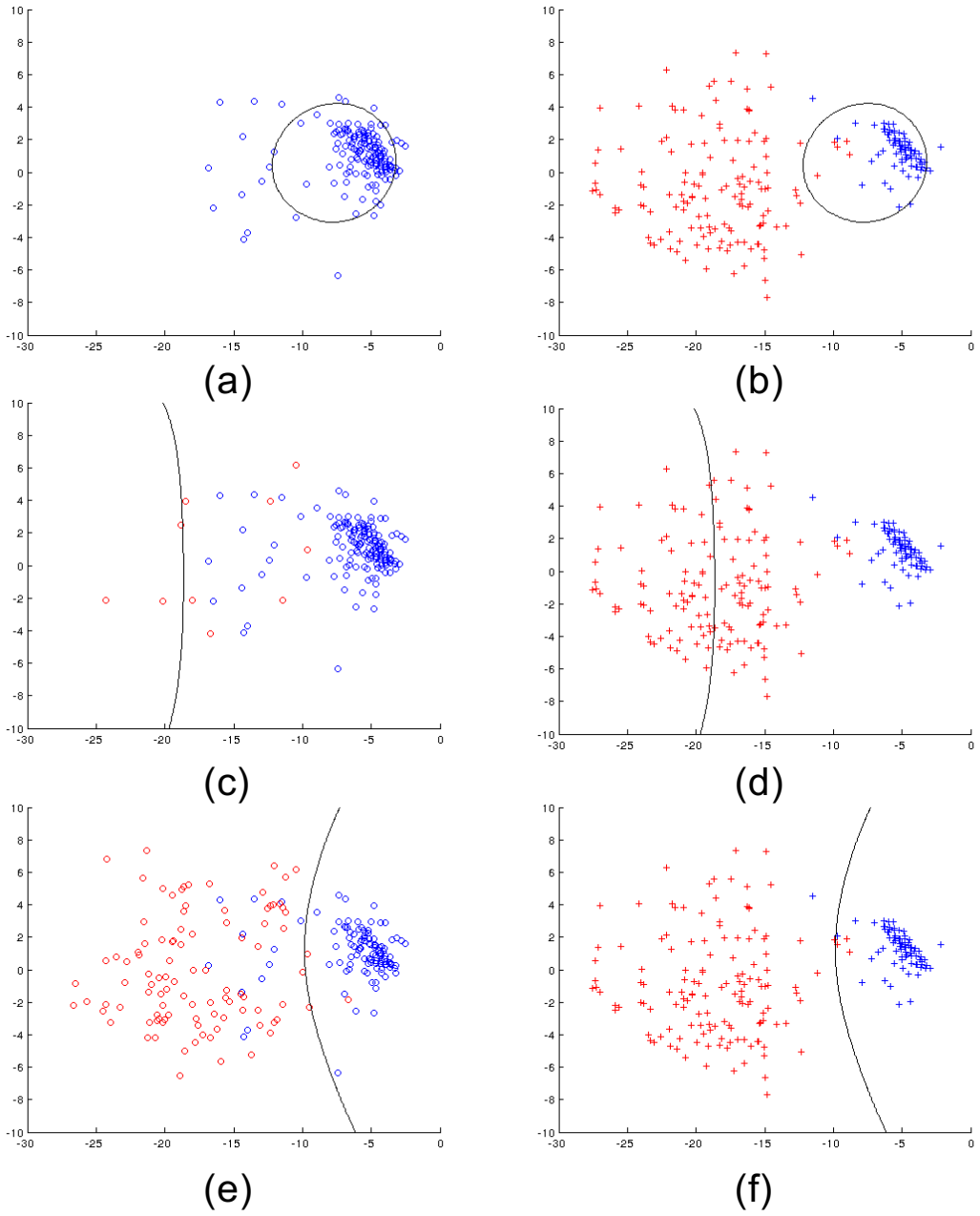

Figure 5: (a) Trained ν-SVMs using 300 benign samples, and (b) its performance on test set; (c)(e) Trained soft-margin SVMs using 290 benign samples plus 10 malignant samples, and 200 benign samples plus 100 malignant samples, respectively, and (d)(f) their performances. Blue: benign samples. Red: malignant samples. Circle: training samples. Cross: test samples. Black curve: decision boundary obtained accordingly. Note when only an insufficient number of malignant samples are available, ν-SVM can find a decision boundary that tightly bounds the benign samples as in (a). However, conventional SVM will be significantly impaired, unless sufficient number of samples from both classes are available as in (e).

problem is

$$\min_{R,\boldsymbol{\xi},c} R^2 + C\sum_{i=1}^{n}\xi_i \tag{10}$$
$$\text{s.t. } \|\Phi(\mathbf{x}_i) - c\|^2 \le R^2 + \xi_i, \xi_i \ge 0,\ i = 1,\ldots,n,$$

where c and R is the center and radius of the desired ball, $\boldsymbol{\xi}$ is the slack variable, and C is a regularization parameter balancing the trade-off between ball radius and data consistency. The dual problem is thus

$$\min_{\boldsymbol{\alpha}} \sum_{i,j=1}^{n}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) - \sum_{i=1}^{n}\alpha_i k(\mathbf{x}_i,\mathbf{x}_i) \tag{11}$$
$$\text{s.t. } 0 \le \alpha_i \le C,\ i = 1,\ldots,n, \text{ and } \sum_{i=1}^{n}\alpha_i = 1.$$

6.2 Relation to ν-SVM

The method of SVDD addresses the same problem in a different way, but interestingly, is closely related to ν-SVM. We describe its relation to ν-SVM be the following theorem.

Theorem 3 (Connection between ν-SVM and SVDD)**.** *If $k(\mathbf{x},\mathbf{y})$ only depends on $\mathbf{x}-\mathbf{y}$, then the solution to ν-SVM is the same with that to SVDD, with $\nu = \frac{1}{nC}$.*

Proof. Firstly, it is obvious that if $k(\mathbf{x},\mathbf{y})$ only depends on $\mathbf{x}-\mathbf{y}$, then $k(\mathbf{x},\mathbf{x})$ will be a constant. If ν is further set to be $\frac{1}{nc}$, then (7) and (11) will be exactly the same problem. Therefore, the optimizing $\boldsymbol{\alpha}$ will be the same for both methods. Then we only need to show that the decision functions of ν-SVM and SVDD coincide given the same $\boldsymbol{\alpha}$.

We already know that

$$\begin{aligned}
\delta_{\nu-SVM}(\mathbf{x}) &= \text{sgn}\left[\sum_i \alpha_i k(\mathbf{x}_i,\mathbf{x}) - \rho\right], \\
\delta_{SVDD}(\mathbf{x}) &= \text{sgn}\left[R^2 - \sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) + 2\sum_i \alpha_i k(\mathbf{x}_i,\mathbf{x}) - k(\mathbf{x},\mathbf{x})\right].
\end{aligned}$$

Let $\mathbf{x}_m$ be one of the points that have $0 < \alpha_m < C = \frac{1}{\nu n}$. Then we have

$$\begin{aligned}
\rho &= \sum_i \alpha_i k(\mathbf{x}_i,\mathbf{x}_m), \\
R^2 &= \sum_{i,j}\alpha_i\alpha_j k(\mathbf{x}_i,\mathbf{x}_j) - 2\sum_i \alpha_i k(\mathbf{x}_i,\mathbf{x}_m) + k(\mathbf{x}_m,\mathbf{x}_m).
\end{aligned}$$

Therefore,

$$\begin{aligned}
\delta_{\nu-SVM}(\mathbf{x}) &= \operatorname{sgn}\left[\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - \sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_m)\right], \\
\delta_{SVDD}(\mathbf{x}) &= \operatorname{sgn}\left[2\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) - 2\sum_i \alpha_i k(\mathbf{x}_i, \mathbf{x}_m) + k(\mathbf{x}_m, \mathbf{x}_m) - k(\mathbf{x}, \mathbf{x})\right].
\end{aligned}$$

Since $k(\mathbf{x}_m, \mathbf{x}_m) = k(\mathbf{x}, \mathbf{x})$ and $\operatorname{sgn}[g(\mathbf{x})] = \operatorname{sgn}[2g(\mathbf{x})]$, we have

$$\delta_{\nu-SVM}(\mathbf{x}) \equiv \delta_{SVDD}(\mathbf{x}).$$

□

Theorem 3 is consistent with our intuition since when $k(\mathbf{x}, \mathbf{y})$ only depends on $\mathbf{x} - \mathbf{y}$, then all the mapped patterns lie on a sphere in the kernel space. Therefore, the smallest sphere found in SVDD can be equivalently segmented by a hyperplane (ν-SVM). It is rather important, not only because it theoretically relates two popular one-class classification methods, but also implies that the generalization error bound derived for ν-SVM also works for SVDD, and some parameter selection methods for SVDD (e.g., [7]) can also be applied to ν-SVM.

7 Conclusion

One-class classification, also known as the data domain description, is not only a classification problem, but also an important step towards learning information and understanding knowledge from training data. The ν-SVM method addresses the one-class classification problem by finding a "smallest" region (the probability density support) that can bound most of the training samples. The resulting optimization problem is similar to that of the conventional SVM, and fast iterative algorithms exist for solving it. Its generalization error has also proved to be bounded from above, which is a very desirable property of learning algorithms.

In this report, we have provided our own proof for the so-called ν-property (Theorem 1), and verified it through a simulation data set. The ν-property provides underlying meaning for the regularization parameter ν, and thus can be leveraged to control the fraction of support vectors and outliers in practice. Real world data are also used to demonstrate the usefulness of ν-SVM when dealing with insufficient negative training samples. Results indicate that when there are insufficient negative samples in the training set, it is better if we only use the positive samples and resort to one-class classification. We have also proved in Theorem 3 that ν-SVM is equivalent to another popular one-class classification method, SVDD, under certain circumstances.

Appendix: Proof of Theorem 2

Before proving Theorem 2, some necessary definitions and lemmas are introduced without proof as follows.

Definition 2 (ϵ-Covering Number)**.** *Let (X, d) is a metric space, and $A \subseteq X$. For $\epsilon > 0$, a set $U \subseteq X$ is called an ϵ-cover for A if for every $a \in A$, $\exists u \in U$ such that $d(a, u) \leq \epsilon$. Then ϵ-covering number of A is the minimal cardinality of an ϵ-cover for A, and is denoted by $\mathcal{N}(\epsilon, A, d)$.*

Specifically in this report, suppose $\mathcal{X}$ is a compact subset of $\mathbb{R}^d$, and $\mathcal{F}$ is a linear function space with the distance defined by the infinite-norm, i.e., for $f \in \mathcal{F}$,$\|f\|_{\ell_\infty} = \max_{\mathbf{x}\in T} |f(\mathbf{x})|$. Then let

$$\mathcal{N}(\epsilon, \mathcal{F}, n) \triangleq \max_{T\in\mathcal{X}^n} \mathcal{N}(\epsilon, \mathcal{F}, \ell_\infty).$$

Definition 3. *Let $L(\mathcal{X})$ be the set of non-negative functions f on $\mathcal{X}$ with countable support. Define 1-norm on $L(\mathcal{X})$ by $\|f\|_1 \triangleq \sum_{\mathbf{x}\in supp(f)} f(\mathbf{x})$. Then*

$$L^B(\mathcal{X}) \triangleq \{f \in L(\mathcal{X}) : \|f\|_1 \leq B\}.$$

Lemma 1 (Theorem 14 in [3])**.** *Suppose we are given a training set $T = \{\mathbf{x}_i\}_{i=1}^n$ generated i.i.d. from an underlying but unknown distribution P which does not contain discrete components, where $\mathbf{x}_i \in \mathcal{X}$, for all i. For any $\gamma > 0$, $f \in \mathcal{F}$, fix $B \geq \mathcal{D}(T, f, \theta)$, then with probability $1 - \delta$*

$$P\{\mathbf{x} : f(\mathbf{x}) < \theta - 2\gamma\} \leq \frac{2}{n}(k + \log_2\frac{n}{\delta}),$$

where $k = \lceil \log_2\mathcal{N}(\gamma/2, \mathcal{F}, 2n) + \log_2\mathcal{N}(\gamma/2, L^B(\mathcal{X}), 2n)\rceil$.

Lemma 2 (Lemma 7.14 of [10])**.** *For all $\gamma > 0$,*

$$\log_2\mathcal{N}(\gamma, L^B(\mathcal{X}), n) \leq b\log_2\frac{e(n + b - 1)}{b},$$

where $b = \lfloor\frac{B}{2\gamma}\rfloor$.

Lemma 3 (Williamson et al. [11])**.** *Let $\mathcal{F}$ be the class of linear classifiers with norm at most 1 confined to a unit ball centered at the origin, then for $\epsilon \geq c/\sqrt{n}$,*

$$\log_2\mathcal{N}(\epsilon, \mathcal{F}, n) \leq \frac{c^2\log_2\left(\frac{2\ln 2}{c^2}\epsilon^2 n\right)}{\epsilon^2},$$

where $c = 103$.

Using Lemma 1, 2, and 3 as tools, we are now ready to prove Theorem 2.

Proof of Theorem 2. Note in Theorem 2,

$$R_{\mathbf{w},\rho-\gamma} = \{\mathbf{x} : f_{\mathbf{w}}(\mathbf{x}) \geq \rho - \gamma\},$$

so we have

$$\{\mathbf{x} : \mathbf{x} \notin R_{\mathbf{w},\rho-\gamma}\} \Leftrightarrow \{\mathbf{x} : f_{\mathbf{w}}(\mathbf{x}) < \rho - \gamma\}.$$

Therefore, the idea is to apply Lemma 1 (by replacing 2γ by γ) for proving Theorem 2.

Firstly, notice that we can treat the offset ρ as 0 without loss of generality. Secondly, in order to invoke Lemma 3 while calculating k in Lemma 1, the linear class $\mathcal{F}$ is required to be confined to a unit ball centered at the origin. Hence, we rescale function f to be $\hat{f} = f_{\mathbf{w}/\|\mathbf{w}\|}$. The decision boundary remains the same if we also rescale $\hat{\gamma} = \gamma/\|\mathbf{w}\|$.

Since in Lemma 1, B is fixed. However, in Theorem 2, B does not have to be fixed. Hence we apply Lemma 1 for each value of

$$\lceil \log_2 \mathcal{N}(\hat{\gamma}/4, L^B(\mathcal{X}), 2n) \rceil. \tag{12}$$

Because for error bound $\frac{2}{n}(k + \log_2 \frac{n}{\delta})$ to be nontrivial, k has to be smaller than $\frac{n}{2}$, so is Eq. (12). Then it suffices to make at most $\frac{n}{2}$ applications, and as a result, uses a confidence of $\frac{\delta}{n/2}$ for each application. Therefore, by Lemma 1, we have

$$P\{\mathbf{x} : \hat{f}(\mathbf{x}) < -\hat{\gamma}\} \leq \frac{2}{n}(k' + \log_2 \frac{n^2}{2\delta}),$$

where

$$k' = \lceil \log_2 \mathcal{N}(\hat{\gamma}/4, \mathcal{F}, 2n) + \lceil \log_2 \mathcal{N}(\hat{\gamma}/4, L^B(\mathcal{X}), 2n) \rceil \rceil.$$

In addtion, letting $b = \left\lfloor \frac{2B}{\hat{\gamma}} \right\rfloor$ and using Lemma 2 and 3, we have

$$\begin{aligned}
k' &\leq \left\lceil \frac{16c^2 \log_2(\frac{\ln 2}{4c^2}\hat{\gamma}^2 n)}{\hat{\gamma}^2} + \left\lceil b \log_2\left(\frac{e(2n+b-1)}{b}\right)\right\rceil \right\rceil \\
&\leq \frac{16c^2 \log_2(\frac{\ln 2}{4c^2}\hat{\gamma}^2 n)}{\hat{\gamma}^2} + b \log_2\left(\frac{e(2n+b-1)}{b}\right) + 2 \\
&\leq \frac{16c^2 \log_2(\frac{\ln 2}{4c^2}\hat{\gamma}^2 n)}{\hat{\gamma}^2} + \frac{2\mathcal{D}}{\hat{\gamma}} \log_2\left(\frac{e(2n+(2\mathcal{D}/\hat{\gamma})-1)}{2\mathcal{D}/\hat{\gamma}}\right) + 2 \\
&= \frac{16c^2 \log_2(\frac{\ln 2}{4c^2}\hat{\gamma}^2 n)}{\hat{\gamma}^2} + \frac{2\mathcal{D}}{\hat{\gamma}} \log_2\left(e\left(\frac{(2n-1)\hat{\gamma}}{\mathcal{D}} + 1\right)\right) + 2 \\
&\triangleq k.
\end{aligned}$$

Then simply by replacing $c_1 = 16c^2$, and $c_2 = \frac{\ln 2}{4c^2}$, Theorem 2 is proved. □

References

[1] P. Moulin and V. V. Veeravalli, "Detection and estimation theory." ECE561 lecture notes, UIUC, 2015.

[2] P. Moulin, "Topics in signal processing: Statistical learning and pattern recognition." ECE544NA lecture notes, UIUC, 2015.

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

[4] J. Platt, "Fast training of support vector machines using sequential minimal optimization," *Advances in kernel methods?support vector learning*, vol. 3, 1999.

[5] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[6] 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`.

[7] D. M. Tax and R. P. Duin, "Uniform object generation for optimizing one-class classifiers," *The Journal of Machine Learning Research*, vol. 2, pp. 155–173, 2002.

[8] W. Wolberg and O. Mangasarian, "Multisurface method of pattern separation for medical diagnosis applied to breast cytology,," in *Proceedings of the National Academy of Sciences*, pp. 9193–9196, Dec 1990.

[9] D. M. Tax and R. P. Duin, "Support vector domain description," *Pattern recognition letters*, vol. 20, no. 11, pp. 1191–1199, 1999.

[10] J. Shawe-Taylor and N. Cristianini, "On the generalization of soft margin algorithms," *Information Theory, IEEE Transactions on*, vol. 48, no. 10, pp. 2721–2735, 2002.

[11] R. C. Williamson, A. J. Smola, and B. Schölkopf, "Entropy numbers of linear function classes.," in *COLT*, pp. 309–319, 2000.