Pattern Classification Using a New Border Identification Paradigm: The Nearest *Border* Technique

Yifeng Li[1,], B. John Oommen[1,1], Alioune Ngom[1], Luis Rueda[1]

[a]*Centre for Molecular Medicine and Therapeutics*
Department of Medical Genetics
University of British Columbia
Vancouver, British Columbia V5Z 4H4, Canada
[b]*School of Computer Science*
Carleton University
Ottawa, Ontario K1S 5B6, Canada
[c]*Department of Information and Communication Technology*
University of Agder in Grimstad, Norway
[d]*School of Computer Science*
University of Windsor
Windsor, Ontario N9B 3P4, Canada

Abstract

There are many paradigms for pattern classification such as the optimal Bayesian, kernel-based methods, inter-class border identification, nearest neighbor methods, nearest centroid methods, among others. As opposed to these, this paper introduces our Nearest *Border* (NB) paradigm (a paradigm that has not been reported in the literature earlier, which we shall refer to as the Nearest *Border* (NB) paradigm). The philosophy for developing such a NB strategy is as follows: Given the training data set for each class, we shall attempt to create borders for each individual class. However, unlike the traditional Border Identification (BI) methods, we do not undertake this by using *inter*-class criteria; rather, we attempt to obtain the border for a specific class in the d-dimensional hyper-space by invoking *only* the properties of the samples *within that class*. Once these borders have been obtained, we advocate that testing is accomplished by assigning the test sample to the class *whose border it lies closest to*. This claim appears counter-intuitive, because unlike the centroid or the median, these border samples are often "outliers" and are, really, the points that represent the class the least. Moreover, inter-class border identification methods intuitively outperform within-class ones. However, we have formally proven this claim, and the theoretical results (for the hyperplane and hypersphere-based one-class classifiers) have been verified by rigorous experimental testing on artificial and real-life data sets. While the solution we propose is distantly related to the reported solutions involving Prototype Reduction Schemes (PRSs) and BI algorithms, it is, most importantly, akin to the recently proposed "*anti*-Bayesian" methods of classification.

Keywords: pattern classification, "*anti*-Bayesian" classification, border identification algorithms, classification using borders, applications of SVMs

1. Introduction

1.1. Overview and Related Fields

The goal of this paper is to present a new paradigm in Pattern Recognition (PR), which we shall refer to as the Nearest *Border* (NB) paradigm. This archetype possesses similarities to many of the well-established methodologies in PR, and can also be seen to include many of *their* salient facets/traits. In order for the reader to capture the intricacies of our contribution, and be

*Corresponding author. A very preliminary and brief version of this paper was presented at AI'13, the 2013 Australasian Conference on Artificial Intelligence, in Dunedin, NZ, in December 2013. The latter version, intended to serve as a claim of the result, did not contain the formal proofs of the assertions, and also included only a brief summary of *some* of the experimental results. The content of this present version is, thus, significantly enhanced.

able to perceive it in the context of the existing state of the art, in this introductory section, we briefly describe some of these methodologies from a *conceptual* perspective.

The problem of classification in machine learning can be quite simply described as follows: If we are given a limited number of training samples, and if the class-conditional distributions are unknown, the task at hand is to predict the class label of a new sample with minimum risk. Within the generative model of computation, one resorts to modelling the prior and class-conditional distributions, and then computing the *a posteriori* distribution after the testing sample arrives. The strength of this strategy is that one obtains an optimal performance if the assumed distribution approximates the actual distribution very well. The limitation, of course, is that it is often difficult, if not impossible, to compute the posterior distribution. The alternative is to work with methods that directly model the latter posterior distribution itself. These methods differ in the approximation of the posterior, such as the Nearest Neighbor (NN) or the k-Nearest Neighbors (k-NN), the Support Vector Machine (SVM) etc. This papers advocates such a philosophy.

The most common challenges that all these techniques encounter are (i) the curse of dimensionality, which is encountered when the dimensionality of the feature space is large, (ii) the *small sample size* scenario encountered when one attempts to obtain a significant performance even though the size of the training set is small, (iii) the *large sample size* scenario, in which the computational resources used are large because of the high cardinality of the training set.

For decades, the NN or k-NN classifiers have been widely-used classification rules. Each class is described using a set of sample prototypes, and the class-identity of an unknown vector is decided based on the identity of the closest neighbor(s), which are found among all the prototypes [**?**]. This rule is simple, and yet it is one of the most efficient classification rules in practice. The application of the classifier, however, often suffers from the higher order of the computational complexity caused by the large number of distance computations, especially as the size of the training set increases in high dimensional problems [**?**], [**?**]. Strategies that have been proposed to solve this dilemma can be summarized into the following categories: (i) reducing the size of the design set without sacrificing the performance, (ii) accelerating the computation by eliminating the necessity of calculating superfluous distances, and (iii) increasing the accuracy of the classifiers designed with the set of limited samples.

A simple strategy for affecting this is, for example, that of: (i) using the mean of the training samples of a class in nearest centroid-like method, (ii) resorting to vector quantization (VQ), and (iii) invoking the Non-Negative Matrix Factorization (NMF) scheme, among others. The strengths of these are that the accuracy may not deteriorate by using only a fewer number of samples or meta-samples, and this can be useful when the data is noisy and/or redundant. One must observe that the testing algorithm is, by definition, faster. The weakness of using a simple parametric strategy, (e.g., the nearest centroid scheme) is that the sample mean merely can not summarize the distribution very well.

The four families of algorithms, which are most closely related to the NB paradigm that we propose, are briefly surveyed below.

Prototype Reduction Schemes: The first of the solutions mentioned above, i.e., of reducing the size of the design set without sacrificing the performance, is the basis for the family of Prototype Reduction Schemes (PRSs), which is central to this paper. The goal here is to reduce the number of training vectors while simultaneously insisting that the classifiers built on the reduced design set perform as well, or nearly as well, as the classifiers built on the original design set. Thus, instead of considering all the training patterns for the classification, a subset of the whole set is selected based on certain criteria. The learning (or training) is then performed on this reduced training set, also called the "reference" set. This idea has been explored for various purposes, and has resulted in the development of many algorithms surveyed in [**? ? ?**]. It is interesting to note that Bezdek *et al.* [**?**], who have composed an excellent survey of the field, report that there are "zillions!" of methods[1] for finding prototypes (see page 1,459 of [**?**]). There are also many *families* of PRSs. In certain families, this reference set not only contains the patterns which are closer to the true discriminant's boundary, but also the patterns from the other regions of the space that can adequately represent the entire training set. While most prototype selection methods use criteria based on the

[1] One of the first of its kind is the Condensed Nearest Neighbor (CNN) rule [**?**]. The CNN, however, includes "interior" samples which can be eliminated completely without changes in the performance. Accordingly, other methods have been proposed successively, such as the Reduced Nearest Neighbor (RNN) rule [**?**], the Prototypes for Nearest Neighbor (PNN) classifiers [**?**], the Selective Nearest Neighbor (SNN) rule [**?**], two modifications of the CNN [**?**], the Edited Nearest Neighbor (ENN) rule [**?**], and the non-parametric data reduction method [**?**]. Additionally, in [**?**], the Vector Quantization (VQ) technique [**?**] was also reported as an extremely effective approach to data reduction. It has also been shown that the SVM can be used as a mean of selecting initial prototype vectors, which are subsequently operated on by LVQ3-type methods [**?**].

full training data, Prototypes can also be selected locally. The clustering-based method proposed in [**?**] is an example of such philosophy.

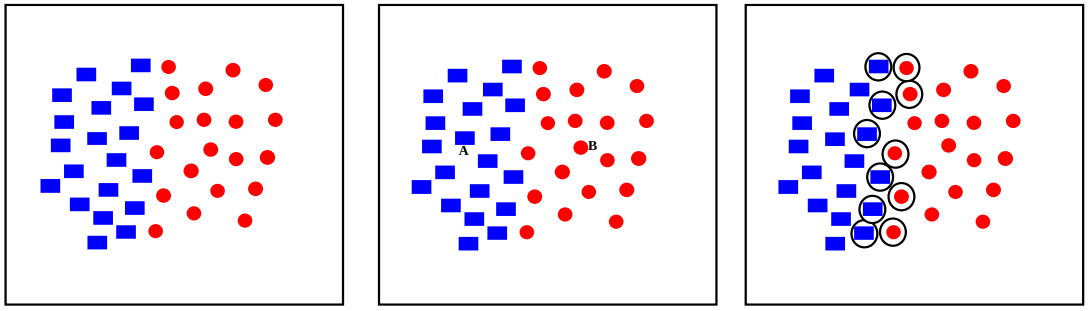
Border Identification Algorithms: Border Identification (BI) algorithms, which are a subset of PRSs, work with a reference set that contains only "border" points. To enable the reader to perceive the difference between general PRSs and BI algorithms, we present some typical data points in Figure **??**. Consider Figure **??** in which the circles belong to class ω_1 and rectangles belong to class ω_2. A PRS would attempt to determine the relevant samples in both the classes which are capable of achieving near-optimal classification. Observe that some samples which fall strictly *within* the collection of points in each class, such as A and B in Figure **??**, could be *Prototypes*, because the testing samples that fall close to them will be correctly classified. As opposed to this, in a BI algorithm, one uses *only* those samples that lie close to the *boundaries* of the two classes, as shown in Figure **??**. In all brevity, we mention that recent research [**?**] has shown that for overseeing the task of achieving the classification, the samples extracted by a BI scheme, and which lie *close to the discriminant function's boundaries*, have significant information when it concerns the power of the classifier. Duch [**?**] and Foody [**?**] proposed algorithms to achieve this. But as the patterns of the reference set described in [**?**] and [**?**] are only the "near" borders, they do not have the potential to represent the entire training set, and hence do not perform well. In order to compete with other classification strategies, it has been shown that we need to also include the set of "far" borders to the reference set [**?**]. A detailed description of traditional BI algorithms namely Duch's approach, Foody's algorithm and the Border Identification in Two Stages can be found in [**?**]. Border identification are often combined with other classification methods, as alternatives to the nearest neighbors. While pairs of border points are used to define class boundaries in [**?**], borders points identified by various methods have also been used to define class centroid, as proposed in [**?**].

SVM-type Algorithms: Representative of a completely distinct family of algorithms is the acclaimed SVM which is known as being quite suitable from a theoretical point of view as well as in practical applications. From the basic theory of the SVM (explained in the Appendix) we know that it makes use of the so-called "sparse" representation, and has the capability of extracting vectors which support the boundary between the two classes, and they can satisfactorily represent the global distribution structure. Also the learning algorithm can be easily expanded to nonlinear problems by employing a technique akin to that of kernel functions. As we shall demonstrate in a subsequent section, our NB paradigm can be implemented by invoking the properties of one-class SVMs.

"Anti-Bayesian" PR Algorithms: A relatively new and distinct paradigm, which works in a counter-intuitive manner, is the recently introduced "anti-Bayesian" philosophy. As a backdrop to this, we mention that when expressions for the *a posteriori* distribution are simplified, the classification criterion that attains the Bayesian optimal lower bound often reduces to testing the sample point using the corresponding distances/norms to the *means* or the "central points" of the distributions. In [**?** **?** **?**], the authors demonstrated that they can obtain optimal results by operating in a diametrically opposite way, i.e., a so-called "anti-Bayesian" manner. They showed that by working with a *few* points *distant* from the mean, one can obtain remarkable classification accuracies. The number of points referred to can be as small as *two* in the uni-dimensional case. Further, if these points are determined by the *quantiles* of the distributions, the accuracy attains the optimal Bayes' bound. They demonstrated that one could work with the symmetric quantiles of the features rather than the distributions of the features themselves [**?** **?** **?**]. It turns out, though, that this process is computationally not any more complex than working with the latter distributions. Alternatively, different from the traditional definition of borders, a new definition of border is proposed in the "Anti-Bayesian" Border Identification (ABBI) method [**?**]. For each class, this method selects a small number of data points that lies neither on the discriminant function's boundary nor too close to the central part of a class distribution.

The state-of-the-art of "Anti-Bayesian" classification is summarized below[2]. Initially, in [**?**], the authors worked with the *quantiles* for the data distributions, and showed how it could achieve near-optimal classification for various uni-dimensional distributions. For uni-dimensional quantile-based PR, their methodology is based on comparing the testing sample with the $(\frac{n-k+1}{n+1})^{th}$ percentile of the first distribution and the $(\frac{k}{n+1})^{th}$ percentile of the second distribution. These results were shown to be applicable for the distributions

[2]In all these papers, the authors had *erroneously* associated the $\frac{n-k+1}{n+1}$ and $\frac{k}{n+1}$ percentiles with the n-order Order Statistics (OS), and in particular, with the $n-k^{th}$ OS of the first distribution and the k^{th} OS of the second. Thus, although the PR schemes reported in [**?**], [**?**] and [**?**] are accurate, they are rather based on the *quantiles* of the distributions and not on the OS. The theoretical results are also true if one views them from the perspectives of the *quantiles* instead of the OSs.



(a) Training Set (b) Prototypes (c) Border Patterns

Figure 1: A schematic view which shows the difference between *Border* patterns and *Prototypes*.

that are members of the symmetric exponential family. By considering the entire spectrum of the possible values of k, the results in [**?**] and [**?**] showed that the specific value of k is usually not so crucial. Subsequently, in [**?**], they proved that these results can also be extended for multi-dimensional distributions.

The challenge involved in using quantile-based criteria is that one needs many training samples to estimate these quantiles. Thus, the question of resolving this for the small sample set is still open.

This brings us to the question of why one needs a new paradigm and what this paradigm entails.

1.2. Problem Formulation

In this paper, we would like to explicitly formulate a paradigm for PR that only uses the "border" points. First of all, the goal is that this process should be independent of the number of dimensions, thus overcoming a handicap inherent in the above-mentioned "Anti-Bayesian" schemes. This would, thus, permit us to apply the BI principle for high-dimensional data. The method that we propose should encapsulate a methodology that is universal for any distribution and should, hopefully, simultaneously crystallize the concept of the border in the multivariate case.

What then does this new paradigm entail? Essentially, we would like it to possess all the salient characteristics of all the four families of methods described above. First and foremost, it should be able to learn the border for each class. To achieve this, unlike the traditional BI methods, we do not resort to using *inter*-class criteria. Rather, we shall compute the border for a specific class in the d-dimensional hyper-space by invoking *only* the properties of the samples *within that class*. Once these borders have been obtained, we advocate that testing is accomplished by assigning the test sample to the class whose border it lies closest to. We claim that this distance is an approximation to the value of the *a posteriori* distribution, which justifies the rule of assigning the testing samples to the nearest border. This claim, appears counter-intuitive, because unlike the centroid or the median, these border samples are often "outliers" and are, indeed, the points that represent the class the least; the inter-class border identification methods are supposed to work better than within-class methods, because the border points are selected in a supervised way in the former methods. The within-class border identification methods are essentially unsupervised; while the inter-class methods are supervised. Using the within-class information only, we do not need to resort to one-versus-one or one-versus-rest scheme for supervised hunting, This is one computational advantage of within-class methods over inter-class ones, especially for many-class data. Furthermore, we also claim that inter-class methods are not necessarily better than within-class ones in terms of accuracy. Of course, being not stereotypic, the integration of both information should improves the performance.

Proposed Solution: Although we state and formalize the nearest-border paradigm from a conceptual perspective, we currently realize it here by applying the *Support Vector Domain Description* (SVDD) for the multi-class problems for which the authors of [**?**] earlier proposed a Bayesian method. First of all, a SVDD representation is learnt for each class. Thereafter, a pseudo-class-conditional-density function is constructed for each class. Finally, the decision is made using the estimated pseudo-posterior probabilities. In this regard, the authors of [**?**] proposed a multi-class classifier by an ensemble of one-class classifiers. First of all, a SVDD or *Kernel Principal Component Analysis-based Kernel Whitening* (KW-KPCA) is applied to each class, where we can see that the SVDD approximates the class boundary by hyper-spheres in the feature space, while the KW-KPCA uses hyper-ellipses. Thereafter, the normalized distance from the prototype of each class is

computed, whence the testing sample is assigned to the class which minimizes this distance. Local SVDD is proposed in [**?**] which locally applies SVDD to describe overlapping regions. These existing methods make use of within-class information only, but do not explicitly crystallize a learning paradigm.

1.3. Contributions of this Paper

The novel contributions of this paper are the following:

- We explicitly and formally propose a new PR paradigm, the Nearest *Border* paradigm, in which we create borders for each individual class, and where testing is accomplished by assigning the test sample to the class whose border it lies closest to.
- Our paradigm falls within the family of PRSs, because it yields a reference set which is a small subset of original training patterns. The testing is achieved by *only* utilizing the latter.
- Our paradigm falls within the family of BI methods, except that unlike traditional BI methods, the borders we obtain do not use *inter*-class criteria; rather, they *only* utilize the properties of the samples *within that class*.
- The Nearest *Border* paradigm is essentially "anti-Bayesian" in its salient characteristics. This is because the testing is not done based on central concepts such as the centroid or the median, but by comparisons using these border samples, which are often "outliers" and which, in one sense, represent the class the least.
- The Nearest *Border* paradigm is closely related to the family of SVMs, because the paradigm can be implemented by *applying* one-SVMs to identify the class borders.
- To justify all these claims, we submit a formal analysis and the results of various experiments which have been performed for many distributions and for many real-life data sets, and the results are clearly conclusive.

We conclude by mentioning that, as far as we know, such a paradigm has not been reported in the PR literature.

1.4. Paper Organization

The rest of the paper is organized as follows. First of all, in Section **??**, we present a fairly comprehensive overview of the NB philosophy. Rather than distract the readers with details, we refer the readers to Appendices **??** and **??** for the brief overview of the foundations of the two-class SVMs, and to a more-detailed study of one-class SVMs that incorporate the hypersphere or hyperplane borders. The paper then continues to the exegesis on NB classifiers in Section **??**. Section **??** details the experimental results obtained by testing our schemes and comparing it with a set of benchmark algorithms. Section **??** concludes the paper.

In the next section, we shall formalize the general theory of the NB classification paradigm.

2. The Theory of NB Classifiers

We assume that we are dealing with a PR problem involving g classes: $\{\omega_1, \cdots, \omega_g\}$. For any specific class ω_i, we define a region $\mathcal{R}_i$ that is described by the function $f_i(\boldsymbol{x}) = 0$ (which we shall refer to as its "border"), where $\mathcal{R}_i = \{\boldsymbol{x} | f_i(\boldsymbol{x}) > 0\}$. We describe $\mathcal{R}_i$ in this manner so that it is able to capture the main mass of the probability distribution $p_i(\boldsymbol{x}) = p(\boldsymbol{x}|\omega_i)$. All points that lie outside of $\mathcal{R}_i$, are said to fall in its "outer" region, $\overline{\mathcal{R}}_i$, where $\overline{\mathcal{R}}_i = \{\boldsymbol{x} | f_i(\boldsymbol{x}) < 0\}$. These points are treated as outliers as far as class ω_i is concerned.

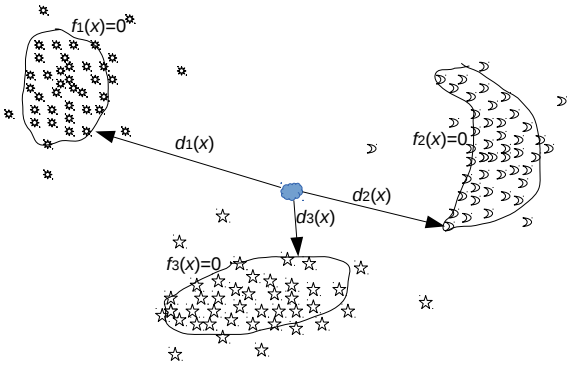
The function $f_i(\boldsymbol{x})$ is crucial to our technique because it explicitly defines the region $\mathcal{R}_i$. Formally, the function $f_i(\boldsymbol{x})$ must be defined in such a way that:

1. $f_i(\boldsymbol{x})$ is the *signed distance* from the point $\boldsymbol{x}$ to the border such that $f_i(\boldsymbol{x}) > 0$ if $\boldsymbol{x} \in \mathcal{R}_i$, and $f_i(\boldsymbol{x}) < 0$ if $\boldsymbol{x} \in \overline{\mathcal{R}}_i$;
2. If $f_i(\boldsymbol{x}_1) > f_i(\boldsymbol{x}_2)$, then $p_i(\boldsymbol{x}_1) > p_i(\boldsymbol{x}_2)$;
3. If $f_i(\boldsymbol{x}) > f_j(\boldsymbol{x})$, then $p(w_i|\boldsymbol{x}) > p(w_j|\boldsymbol{x})$.

In order to predict the class label of a new testing sample $\boldsymbol{x}$, we calculate its signed distance from each class, and thereafter assign it to the class with the minimum distance. In other words, we invoke the softmax rule:

$$j = \arg \max_{i=1}^{g} f_i(\boldsymbol{x}). \tag{1}$$

This idea is illustrated in Figure **??**, where there are three classes: the sun class, moon class, and star class. The training is to learn the border of each class. A new sample, represented by a cloud, is predicted to the star class as its distance to this class is smaller than the other two classes.



The main challenge that we face in formulating, designing and implementing such a NB theory lies in the complexity of conveniently and accurately procuring such borders. The reader will easily see that this is equivalent to the problem of identifying functions $\{f_i(\boldsymbol{x})\}$ that satisfy the above constraints. Although a host of methods to do this are possible, in this paper, we propose one that identifies the boundaries using the one-class SVM[3] described below.

3. Nearest Border Classifiers

Before presenting the rationale and details of the NB classifiers, we feel that it is imperative for the reader to view it from the perspective of two-class and one-class SVMs. In this regard, as mentioned earlier, we present in Appendix **??** and **??**, a brief overview of the *foundations* of the hyperplane and hypersphere-based one-class SVMs. Using the appendices as a backdrop, we now discuss how they can be used to formulate the family of NB classifiers. To do this, we shall first affirm that the two-class SVM actually consists of two hyperplane-based one-class SVMs. Thereafter, we shall present the implementation of the NB paradigm based on the hypersphere-based SVDD.

3.1. One-Class SVM-based Schemes

We shall first state and prove the relationship between the family of hyperplane-based one-class SVMs and the corresponding two-class SVM. This result is given by the following proposition.

Theorem 1. *For two-class data, the task of learning a single two-class SVM is equivalent to that of learning two one-class SVMs under the condition that the hyperplanes of both the one-class SVMs are parallel.*

[3]We are currently investigating an alternate method that involves the α-pruning of the densities. The results that we have are quite exciting, but are rather preliminary.

Proof. Without loss of generality, in our proof, we shall assume that we are considering the case of obtaining the two-class ν-SVM .

Let us suppose that the parallel hyperplanes for the positive and negative classes are:
$f_+(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} - b_+ = 0$, and
$f_-(\boldsymbol{x}) = (-\boldsymbol{w})^{\mathrm{T}}\boldsymbol{x} + b_- = 0$,
where the biases $b_+, b_- > 0$.

With regard to the signs of the respective functions, we mention that:

- $f_+(\boldsymbol{x}) > 0$ if $\boldsymbol{x}$ is on the positive side (the side $\boldsymbol{w}$ pointing to) of $f_+(\boldsymbol{x}) = 0$.
- $f_-(\boldsymbol{x}) > 0$ if $\boldsymbol{x}$ is on the positive side (the side $-\boldsymbol{w}$ pointing to) of $f_-(\boldsymbol{x}) = 0$.

The idea of utilizing one-class classifiers for classification is to maximize the absolute margin between $f_+(\boldsymbol{x}) = 0$ and the origin, as well as the margin between $f_-(\boldsymbol{x}) = 0$ and the origin. In other words, the goal is to maximize *both* $\frac{b_+}{\|\boldsymbol{w}\|_2}$ *and* $\frac{b_-}{\|\boldsymbol{w}\|_2}$.

Now consider the optimization associated with learning of two one-class SVMs with parallel hyperplanes. One can see that this can be formulated as below:

$$\begin{aligned}
\min_{\boldsymbol{w},b_+,b_-,\boldsymbol{\xi}_+,\boldsymbol{\xi}_-} \; & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \boldsymbol{C}^{\mathrm{T}}(\boldsymbol{\xi}_+ + \boldsymbol{\xi}_-) - \nu\frac{b_+ + b_-}{2} \qquad (2)\\
\text{s.t. } & \phi(\boldsymbol{X}_+)^{\mathrm{T}}\boldsymbol{w} - b_+\mathbf{1} + \boldsymbol{\xi}_+ \geq 0\\
& \phi(\boldsymbol{X}_-)^{\mathrm{T}}\boldsymbol{w} - b_-\mathbf{1} + \boldsymbol{\xi}_- \geq 0\\
& \boldsymbol{\xi}_+ \geq 0\\
& \boldsymbol{\xi}_- \geq 0\\
& b_+ > 0\\
& b_- > 0.
\end{aligned}$$

After obtaining the parameters of the model, the hyperplane between the two parallel hyperplanes is $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + \frac{-b_1+b_2}{2} = 0$. If we now re-visit the formulation of the two-class ν-SVM formulation (as given in the previous section), we see that this is:

$$\begin{aligned}
\min_{\boldsymbol{w},b,\rho,\boldsymbol{\xi}} \; & \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \nu\rho + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{\xi} \qquad (3)\\
\text{s.t. } & \phi(\boldsymbol{X}_+)^{\mathrm{T}}\boldsymbol{w} + b\mathbf{1} - \rho\mathbf{1} + \boldsymbol{\xi}_+ \geq 0\\
& -\phi(\boldsymbol{X}_-)^{\mathrm{T}}\boldsymbol{w} - b\mathbf{1} - \rho\mathbf{1} + \boldsymbol{\xi}_- \geq 0\\
& \boldsymbol{\xi}_+ \geq 0\\
& \boldsymbol{\xi}_- \geq 0\\
& \rho \geq 0.
\end{aligned}$$

By a careful examination of the two formulations, one can confirm that we can obtain the *exact same* formulation as in Equation (**??**) by setting $b = \frac{-b_+ + b_-}{2}$ and

$\rho = \frac{b_+ + b_-}{2}$, where $b_+ > 0$ and $b_- > 0$. This concludes the proof. □

Remark: From the above proposition, we can further infer that the two-class SVM is, in fact, an implementation of what we have referred to as the *Nearest Border* paradigm! This is because, whenever we want to assign a new sample, $\boldsymbol{x}$, to a specific class, the SVM decision function:

$$d(\boldsymbol{x}) = \text{sign}[f(\boldsymbol{x})], \tag{4}$$

is equivalent to:

$$j = \arg\max_{i=+/-} f_i(\boldsymbol{x}) = \arg\max_{i=+/-} \frac{f_i(\boldsymbol{x})}{\|\boldsymbol{w}\|}. \tag{5}$$

Further, from the Bayesian learning theory, this formulation is precisely a discriminative model that directly models the *a posteriori* probability distribution.

3.2. The Hypersphere-based Nearest Border *Method*

The nearest centroid approach only uses the means of the class-conditional distribution, and this is the reason why it is not effective for the scenario when the variances of the various classes are very different. As shown above, the two-class SVM can find the boundary of each class, but the solution to this problem cannot be easily and naturally extended to the multi-class problem. The difficulty of extending any linear model from its two-class formulation to its corresponding multi-class formulation, lies in the fact that a hyperplane always partitions the feature space into two "open" subspaces, implying that this can lead to ambiguous regions that may be generated by some extensions of the two-class regions for the multi-class case. The most popular schemes to resolve this are the one-against-rest (using a softmax function) and the one-against-one solutions.

As a one-class model, Tax and Duin's SVDD [**?**] aims to find a closed hypersphere in the feature space that captures the main part of the distribution. By examining the corresponding SVM, we see that the hypersphere obtained by the SVDD is the estimate of feature's *Highest Density Region* (HDR). In particular, for the univariate distribution, the estimation of the *Highest Density Interval* (HDI) is to search for the threshold p^* that satisfies:

$$\int_{x:p(x|D)>p^*} p(x|D)dx = 1 - \alpha. \tag{6}$$

The $(1-\alpha)$% HDI is defined as $C_\alpha(p^*) = \{x : p(x|D) \geq p^*\}$. If we now define the *Central Interval* (CI) by the interval:

$$\begin{aligned} C_\alpha(l,u) = \{x \in (l,u) | & P(l \leq x \leq u|D) \\ & = 1-\alpha, P(x \leq l) = \frac{\alpha}{2}, P(x \geq u) = \frac{\alpha}{2}\}, \end{aligned} \tag{7}$$

one will see that, for symmetric unimodal univariate distribution, HDI coincides with the CI. However, for non-symmetric univariate distributions, the HDI is smaller than the CI.

For known distributions, the CI can be estimated by the corresponding quantile. However, for unknown distributions, the CI can be estimated by Monte Carlo approximation. However, in the context of this paper we remark that by virtue of Vapnik's principle, it is not necessary to estimate the density by invoking a non-parametric method.

For multivariate distributions, we can estimate the $(1-\alpha)$% HDR $C_\alpha(f)$ by using the equality:

$$\min_f \int_{f(x)\geq 0} 1dx, \text{ s.t. } \int_{x:f(x)\geq 0} p(x|D)dx = 1-\alpha. \tag{8}$$

We shall refer to this optimal contour $f^*(x) = 0$ as the $(1-\alpha)$-border/contour.

Our idea for classification is in the following: We can learn a hypersphere for each class in the feature space in order to describe the border of this class. We then calculate the distance from a unknown sample to the border of each class and assign it into the class with the minimum distance. The training phase of our approach is to learn the hypersphere $f_i(\boldsymbol{x}) = 0$ parameterized by $(\boldsymbol{c}_i, R_i)$ for each class as specified by Equation (**??**). The prediction phase then involving assigning the unknown sample $\boldsymbol{x}$ using the following rule:

$$j = \arg\max_{i=1}^{g} f_i(\boldsymbol{x}), \tag{9}$$

where $f_i(\boldsymbol{x})$ is defined as in Equation (**??**). In particular, we note that:

- $f_i(\boldsymbol{x}) \in \mathbb{R}$ is the signed distance of $\boldsymbol{x}$ from the corresponding boundary;
- For points inside the i-th hypersphere, $f_i(\boldsymbol{x}) > 0$;
- For points outside the hypersphere, $f_i(\boldsymbol{x}) < 0$. Further, the larger $f_i(\boldsymbol{x})$ is, the closer it is to class ω_i, and the higher the value of $p(w_i|\boldsymbol{x})$ is. From the parameters of $f_i(\boldsymbol{x})$, we can see that $f_i(\boldsymbol{x})$ considers both mean and variance of the distribution. It can be further enhanced by the *normalized distance* through the operation of dividing it by R_i.

This, quite simply, leads us to the following decision rule:

$$j = \arg\max_{i=1}^{g} \frac{f_i(\boldsymbol{x})}{R_i}. \quad (10)$$

We refer to this approach above as the *Nearest Border approach based on HyperSphere* (NB-HS).

In an analogous manner, the two-class SVM can also be called the *Nearest Border approach based on HyperPlane* (NB-HP). The advantage of using the (normalized) distance from the border instead of the mean as in nearest centroid approach is that the former takes into account both the means and the variances, while the later only considers the mean. The advantage of the NB-HS over the SVM is that, due to the closure property of the hypersphere, the borders obtained in the NB-HS can be estimated one-by-one which, is more computationally efficient than by invoking a one-against-rest SVM. Hereafter, the hypersphere based NB using the decision rule specified by Equation (**??**) will be denoted by *ν-NB*, and the one that utilizes the normalized distance, as in in Equation (**??**) will be denoted by *ν-NBN*.

As mentioned in Section **??**, ν is the upper bound of the fraction of outliers and the lower bound of the fraction of the support vectors[4]. As the number training samples increases to infinity, these two bounds converge to ν. However, in practice, we usually have a very limited number of training samples. In order to obtain ν which corresponds to the α fraction of outliers, firstly, we need to let $\nu = \alpha$, and then reduce ν gradually until the α fraction of outliers are obtained. This variant of NB will be named the *α-NB* in the subsequent sections.

The dual form of one-class SVM, formulated in Equation (**??**), is a constrained quadratic programming. Its computational complexity depends on the number of training samples in a class, rather than the number of features. It thus makes the classification of high-dimensional data (for example text and image data) very efficient. Through the last decade, various methods have been proposed to solving such large-scale quadratic programming. For example the SMO algorithm mentioned above takes linear steps to until convergence. In the situation of a huge number of classes, a computational benefit of using merely within-class information instead of inter-class information is that a vast number of inter-class comparisons can be avoided, even though a gain of classification accuracy is expected when considering inter-class discrimination.

3.3. Relationships with Existing Paradigms

It is also prudent for us to clarify the relationship between this newly-introduced NB paradigm and the four schemes mentioned in Section **??**. All these methods endeavor to obtain a reference set of data points that can characterize the distribution of data. The NB paradigm belongs to the family of BI algorithms, but yields the border points by merely utilizing the information contained in the "*within*-class" points. Furthermore, the way by which the NB scheme classifies a new sample is distinct from the way the family of BI schemes does this. In the NB, the border of each class can be estimated by (but not limited to) invoking the properties of one-class SVMs. Indeed, other alternative implementations of NB classifiers are discussed in Section **??**. It is also pertinent to mention that our NB solution extends, in one sense, the quantile-based anti-Bayesian method in a multi-dimensional context that was not explored before.

4. Experimental Results

The NB schemes that we introduce in this paper have been rigorously tested. In this section, we present a summary of the experiments done and the corresponding results. Our computational experiments can be divided into two segments. First of all, we investigated the performance of our method on three artificial data sets. Subsequently, we statistically compared our approach with benchmark classifiers on 17 well-known real-life data sets. The methods that we have used and the benchmark methods are listed in Table **??**. These methods have not been chosen randomly or haphazardly. The methods, which include the "anti-Bayesian" border identification method that considers inter-class information, naive Bayes[5], the nearest neighbor, nearest centroid, nearest subspace and the SVM, are all prototypes of well-established classical pattern recognition paradigms, and are also philosophically related, in one sense, to to our NB paradigm.

Before we explain the experimental results we would like to emphasize the fact that we are not attempting to

[4] Elsewhere, some of the authors of this paper have succeeded in designing a *Sequential Minimal Optimization* (SMO) algorithm to solve Equation (**??**). The SMO is an extreme case of the decomposition method [**?**]. Its principle lies in the fact that: It works iteratively until the KKT conditions are satisfied. In each iteration, two of the set of working variables are selected by a heuristic. Thereafter, we determine if at least one of these variables violates the KKT conditions. Then, these two working variables are updated analytically and the rest are kept fixed.

[5] The naive Bayes classifier "crashed" on some real-life data sets, and thus, in the interest of fairness, its results on real-life data sets have not been included.

demonstrate that our new technique is the "best available" scheme. Rather, our intention is to show that such a NB strategy is not only feasible – it is also extremely competitive, yielding an accuracy which is close to the best reported PR methodologies. Indeed, in some cases, its accuracy even exceeds the accuracy of the SVM. Surprisingly, the NB methods using merely within-class information to identify borders generally outperform the inter-class methods.

4.1. Accuracy on Synthetic Data

In order to investigate the behaviour of the NB models in various situations, we tested our approaches on three different synthetic data sets described as follows.

1. First of all, in order to compare the performance of our NB approaches with existing ones in the homoscedastic case, we generated four two-dimensional normally distributed classes. These classes had the same standard deviation in each dimension ($\sigma = 1$) but possessed different means. Each class contains 100 data points. This data set has been denoted by *SameVar*, and is illustrated in Figure **??**.
2. Secondly, in order to compare the NB approaches with the NC in the case when the classes had different variances, we generated four Gaussian classes using different variances. We denote this data set by *DiffVar*, and this is shown in Figure **??**.
3. Thirdly, to test the performance of the classifiers for nonlinear scenarios, we used the data set referred to as *NonLinear*, and is shown in Figure **??**. Here we used a Laplacian noise ($\mu = 0, \sigma = 0.15$), which was added to each point.

For the artificial data sets, we compared our approaches with the ABBI, Naive Bayes, NN, NC, and SVM classifiers. The linear kernel was used for our methods, the NC, and SVM approaches on the first two data sets because we wanted to compare them in the input space, and the *Radial Basis Function* (RBF) kernel was used on the last data set because we wanted to compare them in the appropriate feature space. On all three data sets, we used our multi-class ABBI method that were extended by a one-versus-one scheme. For each pair of classes, the number of border points in each class ranges from 5 to 15, and the number of nearest neighbours were searched in $\{1, 3, 5, 7\}$. With regard to the testing strategy, we ran a 3-fold cross-validation on each data for 20 times. All the classifiers used the same training and testing splits in order to maintain a fair comparison. From the 20 results, we computed the mean and

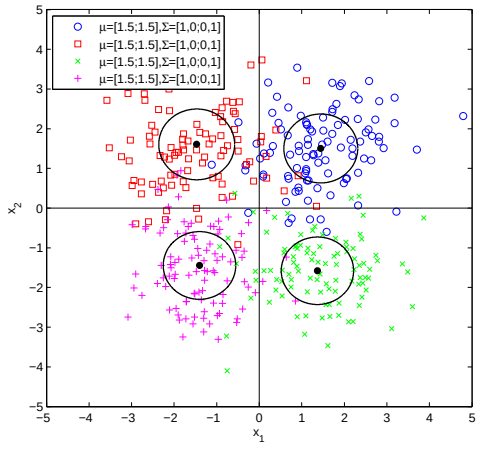


Figure 2: Plot of the *SameVar* data set. In every class, the black dot and circle are the center and the border learned by SVDD respectively.

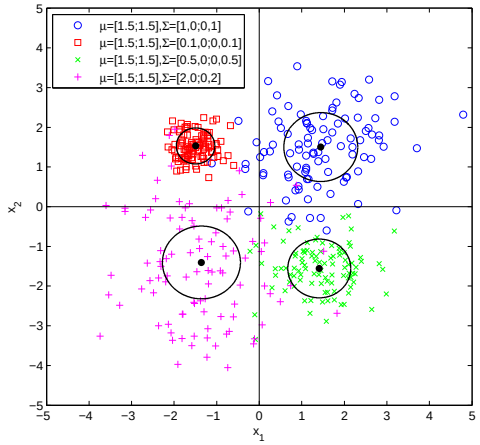


Figure 3: Plot of the *DiffVar* data set. In every class, the black dot and circle are the center and the border learned by SVDD respectively.

standard deviation (STD) of the accuracies, and the results are illustrated in Figure **??**.

On the *SameVar* data set, firstly, we can see that there is no significant difference between the ν-NB and ν-NBN, and α-NB. All of them yielded an almost-equivalent accuracy as the Naive Bayes. Secondly, it can be seen from Figure **??** that the NB was able to identify the centers of each class accurately. The borders have the same volume, which demonstrates that the NB can identify the borders consistent with the variances. The NB approaches yielded an accuracy similar to the NC, which is reasonable because the identical variance of all classes is of no consequence to the NB. Thirdly, although ABBI considered inter-class informa-

Table 1: Summary of our NB methods and the Benchmark methods used.

Category	**Method**	**Description**
Proposed	ν-NB	ν is the lower bound of fraction of SVs and upper bound of the fraction of error. Here, we invoke the decision rule specified by Equation (**??**).
	ν-NBN	Here, ν-NB uses the normalized distance as defined by Equation (**??**).
	α-NB	Here α is the fraction of SVs, and we invoke the decision rule specified by Equation (**??**).
Inter-class BI	ABBI	The "anti-Bayesian" border identification method [**?**] redefines the concept of borders and takes inter-class information into account. The Mahalanobis and Euclidean distance metrics were employed for low and high dimensional data, respectively. For multi-class data, ABBI was extended by a one-versus-one scheme.
Generative	Naive Bayes	This rule has only been used on artificial data. It may fail on real data.
Discriminative	NN	This is the Nearest Neighbor rule [**?**]. Here, we replace the inner product in the Euclidean distance with the RBF kernel, since the latter does not change the NN. Thus, we have invoked the kernelized NN rule.
	NC	This is the Nearest Centroid (or prototype) [**?**] rule. Again, we extended it to the kernelized version.
	NS	This is a Nearest Subspace method proposed in [**?**] (originally called the *Linear Regression Classifier*). Since this method only works safely under the condition that the number of features must be greater than the class-sample-size, we again extended it into the kernelized version in order to let it operate under all conditions.
	SVM	In this case, we used the ν-SVM [**?**], where the one-versus-rest scheme and softmax function are used for the multi-class task.

Table 2: Results of the accuracies achieved by a 3-fold cross-validation using the new and benchmark algorithms on the artificial data sets.

Method	sameVar	diffVar	nonlinear
ν-NB	0.8716(0.0078)	0.9170(0.0056	0.9788(0.0049)
ν-NBN	0.8751(0.0048)	0.9107(0.0057)	0.9749(0.0054)
α-NB	0.8753(0.0065)	0.9175(0.0060)	0.9791(0.0048)
ABBI	0.8515(0.0107)	0.9136(0.0075)	0.9184(0.0174)
Naive Bayes	0.8764(0.0038)	0.9314(0.0039)	0.9264(0.0165)
NN	0.8121(0.0147)	0.8929(0.0086)	0.9818(0.0037)
NC	0.8738(0.0033)	0.8959(0.0027)	0.9408(0.0118)
SVM	0.7765(0.0238)	0.8430(0.0270)	0.9881(0.0031)

tion, it only obtained medium result. Thus, we can say that the within-class paradigm is not necessarily inferior to the inter-class one. Finally, the NN and SVM do not obtain comparable results. This is because the distance measure of the NN is affected by noise, and the SVM is not able to "disentangle" each class well using a one-versus-rest scheme.

On the *DiffVar* data set, first of all, we see that the results again confirm that the NB can identify the borders consistent with the variances (see Figure **??**). The mean accuracies of all the NB approaches and ABBI were very close to the Naive Bayes classifier. However, the NC yielded a worse result than the NB. This is because the variance information helped the NB, while the NC scheme did not consider it.

Finally, for the *NonLinear* data set, firstly, we affirm that all our NB methods and the SVM yielded comparably good results. Secondly, the Naive Bayes did not work well this time, because the data was not Gaussian. Further, the kernel NC was not competent either, because the data in the high-dimensional feature space may have different variances for all the classes. The accuracy of ABBI is not comparable with the NB methods. Since the class distributions are not convex, a small number of border points identified by ABBI cannot be sufficient to represent the boundaries. However,

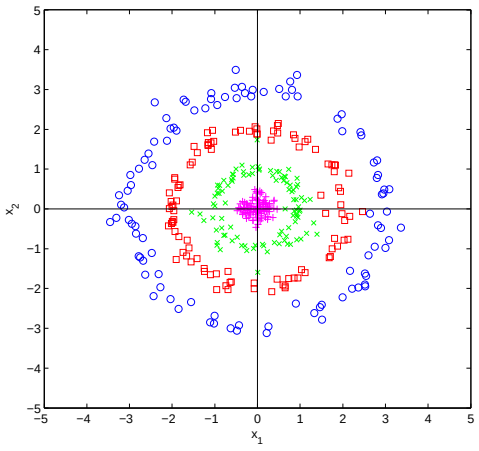


Figure 4: Plot of the *NonLinear* data set.

we think the performance of ABBI in this situation can be improved by kernel techniques.

4.2. Accuracy on Real-Life Data

In order to fully demonstrate the performance of our NB approaches, we also compared them with benchmark approaches on 17 various data sets from bioinformatics, face recognition, hand digits recognition, speech recognition, and so on. These data sets are summarized in Table **??**.

Table 3: The real-life data sets used in our experiments.

Data	#Class	#Feature	#Sample
DNA [**?**]	3	180	2000
ExYaleB[**?**]	38	32256	2432
Ionosphere [**?**]	2	34	351
Iris [**?**]	3	4	150
Letter [**?**]	26	16	15000
MFEAT [**?**]	10	649	2000
Minsteries [**?**]	2	400	326
Pendigit [**?**]	10	16	10992
Pima [**?**]	2	8	768
Satimage [**?**]	6	36	4435
Segment [**?**]	7	19	2310
Svmguide2 [**?**]	3	20	391
Svmguide4 [**?**]	3	20	391
USPS [**?**]	10	256	9299
Vehicle [**?**]	4	18	846
Vowel [**?**]	11	10	990
Wave2 [**?**]	3	40	5000

Methods and Parameters: In this set of experiments, we included the ν-NB and the ν-NBN in the competition. However, we did not involve the α-NB on the real-life data sets, because it would have yielded the same performance as the ν-NB when the parameter (ν in ν-NB or α in the α-NB) is selected by inner 3-fold cross-validation on the training set. The benchmark methods included the ABBI, NN, NC, NS, and the SVM. In this set of tests involving real-life data, we did not include the Naive Bayes classifier because it failed on some of them. Again, we used the RBF kernel in our schemes and in all the benchmark classifiers except ABBI which applied the recommended Mahalanobis or Euclidean distance. All the parameters in each method were selected by a grid or a line-search based on the inner 3-fold cross-validation accuracy of the training set. For ν-NB and ν-NBN, the range of ν was tested from the range $\max(0.025, \frac{1}{\frac{2}{3}s})$ to 0.95 by using a step-size of 0.025, where s was the mean class-sample-size of the training set. For the ν-SVM, the range of ν was from $\max(0.025, \frac{1}{\frac{2}{3}s})$ to $\min(f, 0.95)$, where f was the maximum feasible value of ν defined in [**?**]. For NC, NS and SVM, the parameter σ was searched for from 2^{d-2} to 2^{d+2} by involving a step-size 0.5 in the power, where $d = \log_2(\sqrt{m})$ (where m is the number of features). This was inspired by LIBSVM [**?**] which sets the default value of σ to be 2^d. The parametric setting of ABBI was the same as on the synthetic data.

The results of the accuracies of achieving a 3-fold cross-validation using the new and benchmark algorithms on the real-life data sets are given in Table **??** and plotted in Figure **??**. The results that we achieved in this case, seem to categorically demonstrate the power of the scheme. All the three NB algorithms are almost always better that all the other benchmark algorithms, except the SVM. This is not too difficult to understand because the SVM utilizes the information gleaned by invoking the borders from both the classes. As opposed to this, the NB border merely concentrates on the border that the testing sample is nearest to. The crucial issue that these results communicate is the fact that the NB strategy that we have proposed is a viable and competitive solution, and lends credibility to the fact that the new concept that one can use "borders" (or outliers) to achieve very accurate and almost-optimal PR.

Interpretation of the Results: With regard to the interpretation of the results, we state:

- First of all, as can be seen from the results, the difference between the ν-NB and the ν-NBN is negligible. However, ν-NB has a marginally higher rank than the ν-NBN. Therefore, we can state that using an enhanced distance measure, as defined in Equation (**??**), is beneficial.

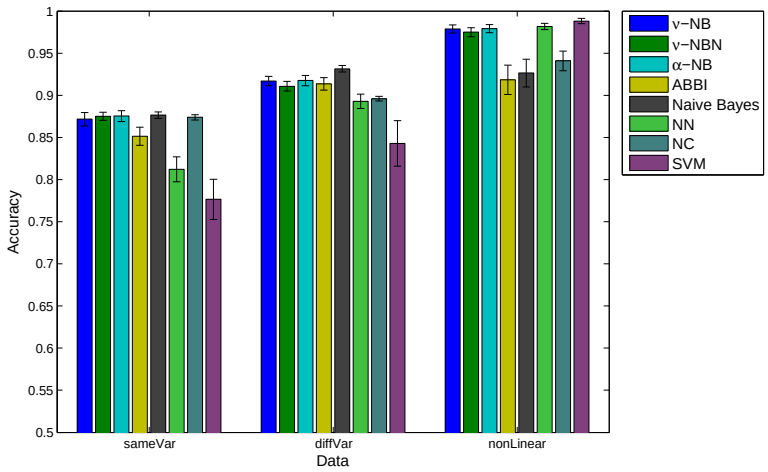


Figure 5: Mean accuracy and STD on the three synthetic data sets.

Table 4: Results of the accuracies achieved by a 3-fold cross-validation using the new and benchmark algorithms on the real-life data sets.

Data	ν-NB	ν-NBN	ABBI	NN	NC	NS	SVM
DNA	0.7955	0.7635	0.6075	0.6990	0.8915	0.4525	0.9385
EYaleB	0.7430	0.7364	0.5831	0.7455	0.0259	0.0263	0.9239
Ionosphere	0.8632	0.8746	0.7236	0.8604	0.7920	0.8063	0.9402
Iris	0.9267	0.9067	0.9400	0.9333	0.8733	0.7000	0.9467
Letter	0.9248	0.9245	0.7743	0.9352	0.7209	0.0517	0.9157
MFEAT	0.9640	0.9640	0.8505	0.9800	0.9455	0.5200	0.9745
Minsteries	0.6258	0.6595	0.6472	0.6043	0.4509	0.6472	0.7454
Pendigits	0.9829	0.9823	0.9385	0.9929	0.8686	0.9925	0.9944
Pima	0.7227	0.7240	0.6484	0.6810	0.7344	0.7005	0.7578
Satimage	0.8638	0.8634	0.8408	0.8970	0.7932	0.9042	0.8992
Segment	0.9065	0.9065	0.8892	0.9558	0.8476	0.2069	0.9468
Svmguide2	0.7877	0.7852	0.7596	0.7161	0.7903	0.7212	0.8031
Svmguide4	0.6601	0.6405	0.3137	0.6618	0.5376	0.3644	0.7598
USPS	0.8635	0.8621	0.8334	0.9525	0.1167	0.6698	0.9505
Vehicle	0.7139	0.7128	0.5236	0.6950	0.5816	0.2388	0.8002
Vowel	0.9434	0.9434	0.9071	0.9646	0.8182	0.9697	0.9455
Wave2	0.8492	0.8484	0.6998	0.7222	0.8086	0.6712	0.8538

- Secondly, the SVM obtained the highest rank. However, by using Friedman test [**?**], there is no significant difference among between the SVM, the NN, and the ν-NB under the significant level of 0.05. This is quite a remarkable conclusion.

- Thirdly, the ABBI method generally had an inferior performance than our new methods. It is apparently surprising, because the ABBI considers the inter-class information, but the NB not. However, we should understand that, working in the input space, ABBI may not be able to select good border points for distorted class distributions. The kernel extension of ABBI, which we are working on, may improve the accuracy.

- Furthermore, the underperformed results of NC and NS are very close to each other.

- Lastly, if we examine the accuracies of the classifiers, we can clearly identify two distinct groups: {SVM, NN, ν-NB, ν-NBN}, and {ABBI, NC, NS}, demonstrating that our newly-introduced NB schemes are competitive to the best reported algorithms in the literature.

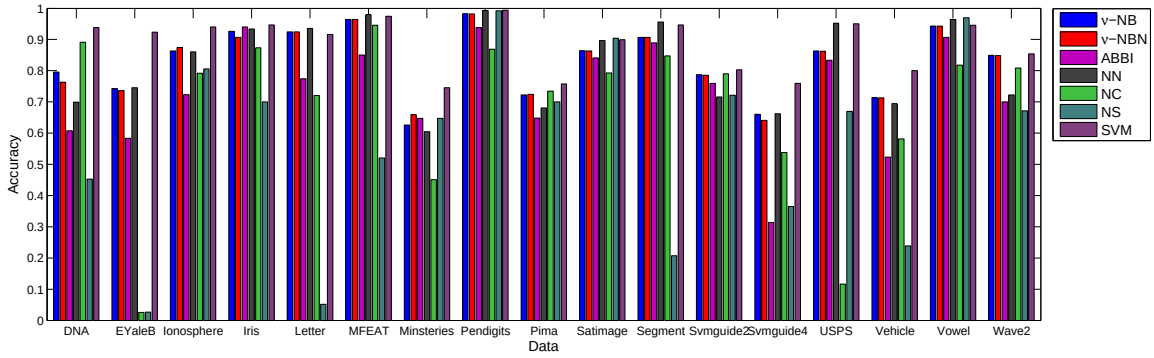


Figure 6: The accuracies achieved on 3-fold cross-validation for the 17 real-life data sets.

5. Conclusions and Future Work

We have introduced a new paradigm for Pattern Recognition (PR) which has not been formally or explicitly investigated in the literature earlier, which we shall refer to as the Nearest *Border* (NB) paradigm. This paradigm can be contrasted with the reported and existing PR paradigms such as the optimal Bayesian, kernel-based methods, inter-class border identification, nearest neighbor methods, nearest centroid methods, among others. The philosophy for developing such a NB strategy is also quite distinct from the above methods (what has been used in the existing literature), because we shall attempt to create borders for each individual class *only* from the training data sets of *that class*. Indeed, unlike the traditional Border Identification (BI) methods, we have not achieved this by using *inter*-class criteria, but by searching for the border for a specific class in the d-dimensional hyper-space by invoking *only* the properties of the samples *within that class*. This has been, in turn, achieved, using the corresponding one-class SVM-based classifers. Once these borders have been obtained, we advocate that testing is accomplished by assigning the test sample to the class whose border it lies closest to. We emphasize that our methodology is actually counter-intuitive, because unlike the centroid or the median, these border samples are often "outliers" and are, indeed, the points that represent the class the least.

We implemented the NB methods, ABBI algorithm, two-class and one-class SVMs in MATLAB. The source code is publicly available our the Regularized Linear Models and Kernels Toolbox [**?**].

The paper has rigorously derived the one-class classifiers for the hyperplane and hypersphere-based schemes, and the theoretical results have been verified by rigorous experimental testing on artificial and 17 real-life data sets. While the solution we propose is distantly related to the reported solutions involving Prototype Reduction Schemes (PRSs) and BI algorithms, it is, most importantly, akin to the recently proposed "*anti*-Bayesian" method that involve the quantiles of the various distributions.

Even though we, in this paper, apply one-class SVMs to identify the borders of the classes, we believe there are many other alternatives. For example, we can identify the contours of a distribution by "taking off" the largest convex hulls constructed from the training data points. Another possibility would be that of using the data points that are furthest from the centers of the masses, to estimate the borders. While the concept of the NB paradigm is broad, we also admit that our current implementation of relying on one-class SVMs can be improved, because the success of the one-class SVM is based on the assumption of dealing with unimodal distributions. We believe, though that we can address this limitation for multi-modal distributions by invoking a good clustering method (for example, NMF [**?**]) to partition any given class into a set of subclasses, and thereafter utilizing a BI method for each subclass. Also, while our current implementations and the above alternatives are unsupervised when learning the border of each class, we believe that we can improve it by integrating the within-class and inter-class information, where the challenge is how to deal with the inter-class information without using a one-versus-one or one-versus-rest scheme. This challenge is more crucial for many-class data.

Acknowledgements

The authors are grateful for the partial support provided by NSERC, the Natural Sciences and Engineering

Research Council of Canada, and OGS, Ontario Graduate Scholarship. Y. Li also appreciate the resources provided by Dr. Wyeth Wasserman so that Y. Li can continue the research in the area of machine learning.

Appendix A. Two-Class SVMs

Since the formulation and analysis of the SVM is fundamental to our technique, a brief overview of its mathematical foundations is not out of place, because without it the process of formulating the specific one-class boundaries is not easily understood.

The linear model for the classification of two-class data is to learn the parameters of the following model:

$$f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b, \tag{A.1}$$

where $\boldsymbol{w}$ is normal vector to the hyperplane, and b is the bias. The decision function is the indicator:

$$d(\boldsymbol{x}) = \mathrm{sign}[\, f(\boldsymbol{x}|\boldsymbol{w}^*, b^*)], \tag{A.2}$$

where $\{\boldsymbol{w}^*, b^*\}$ is the optimal parameter with respect to some criteria. Maximum-margin linear models can be generally expressed by the following formula:

$$\min_{\boldsymbol{w},b} \sum_{i}^{n} l(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b, y_i) + \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2, \tag{A.3}$$

where $l(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b, y_i)$ is a loss function, and λ controls the trade-off between the approximation error and model complexity.

The standard SVM applies the so-called "Hinge" loss: $l(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b, y_i) = \max(0, 1 - y_i(\boldsymbol{w}^{\mathrm{T}}\boldsymbol{x}_i + b))$. The geometric interpretation of the standard SVM is that the margin between two classes is maximized while keeping the samples of the same class at one side of the margin. It is also equivalent to find two closest points of the (reduced) convex hulls, where each class defines a convex hull, and the two closest points determine the separating hyperplane [**?**]. For notational convenience, we define the margin border close to the positive class to be *positive margin border*, and the one close to the negative class to be *negative margin border*. We can also represent the final solution in the form of inner products, so that their corresponding kernel extensions can be easily reached.

In the following, we shall first introduce the C-SVM and the equivalent ν-SVM. With regard to notation, we shall represent the training data set by the matrix $\boldsymbol{X} \in \mathbb{R}^{m\times n}$ where each column corresponds to a training sample. The class labels are in the column vector $\boldsymbol{y} \in \{-1, +1\}^n$.

Appendix A.1. C-SVM

The soft-margin SVM attempts to maximize the margin and simultaneously minimize the relaxation. Consequently, the optimization task of the soft-margin SVM can be expressed by the equation:

$$\begin{aligned} &\min_{\boldsymbol{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{\xi} \\ &\text{s.t. } \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} + b\boldsymbol{y} \geq \boldsymbol{1} - \boldsymbol{\xi} \\ &\quad \boldsymbol{\xi} \geq 0, \end{aligned} \tag{A.4}$$

where $\boldsymbol{Z}$ is sign-changed training samples with its i-th column defined as the element-wise multiplication of the class label and the input vector of the i-th training sample, that is $\boldsymbol{z}_i = y_i * \boldsymbol{x}_i$. $\boldsymbol{\xi}$ is a vector of slack variables. $\boldsymbol{C} = \{C\}^n$ is a parameter balancing the model complexity and loss.

By considering the Lagrangian function for this optimization, the *Karush-Kuhn-Tucker* (KKT) conditions, we can have the dual form of the optimization:

$$\begin{aligned} &\min_{\boldsymbol{\mu}} g(\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\boldsymbol{\mu} - \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{1} \\ &\text{s.t. } \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{y} = 0 \\ &\quad 0 \leq \boldsymbol{\mu} \leq \boldsymbol{C}. \end{aligned} \tag{A.5}$$

One can show that the normal vector is a non-negative linear combination of the training samples, that is:

$$\boldsymbol{w} = \boldsymbol{Z}\boldsymbol{\mu} = \boldsymbol{X}(\boldsymbol{\mu} * \boldsymbol{y}) = \boldsymbol{X}_{\mathcal{S}}(\boldsymbol{\mu}_{\mathcal{S}} * \boldsymbol{y}_{\mathcal{S}}), \tag{A.6}$$

where $\mathcal{S}$ is the set of indices of non-zero multipliers: $\mathcal{S} = \{i|\mu_i > 0, i = 1, \cdots, n\}$. The training samples corresponding to $\mathcal{S}$ are called the *Support Vectors* (SVs), as they are either on the correct margin border or at the wrong side of the correct margin border. In order to compute the bias b, we need to find some points on the boundary, denoted by $\boldsymbol{X}_{\mathcal{B}}$, where $\mathcal{B} = \{i|0 < \mu_i < C, i = 1, \cdots, n\}$. By solving this we obtain $b = \frac{\boldsymbol{y}_{\mathcal{B}} - \boldsymbol{X}_{\mathcal{B}}^{\mathrm{T}}\boldsymbol{w}}{|\mathcal{B}|} = \frac{\boldsymbol{y}_{\mathcal{B}} - \boldsymbol{X}_{\mathcal{B}}^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}(\boldsymbol{\mu}_{\mathcal{S}} * \boldsymbol{y}_{\mathcal{S}})}{|\mathcal{B}|}$.

After obtaining the optimal $\boldsymbol{w}$ and b, the linear function used by the decision function can be computed as follows:

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}(\boldsymbol{\mu}_{\mathcal{S}} * \boldsymbol{y}_{\mathcal{S}}) + \frac{\boldsymbol{y}_{\mathcal{B}} - \boldsymbol{X}_{\mathcal{B}}^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}(\boldsymbol{\mu}_{\mathcal{S}} * \boldsymbol{y}_{\mathcal{S}})}{|\mathcal{B}|}. \end{aligned} \tag{A.7}$$

As per the KKT conditions, we can obtain the following important geometric interpretations from the optimal multipliers. (1) If $\mu_i > 0$, the training point $\boldsymbol{x}_i$ resides either on or outside its correct margin border. (2)

If $0 < \mu_i < C$, $\boldsymbol{x}_i$ is on the margin border. (3) If $\boldsymbol{x}_i$ is on the wrong side of the corresponding margin border, $\mu_i = C$. However, the reverse is not always true. If $\mu_i = C$, $\boldsymbol{x}_i$ is either on or outside the correct margin border.

Appendix A.2. ν-SVM

Suppose that $\boldsymbol{x}$ is a point located on its correct border of the margin. Its corresponding value of $f(\boldsymbol{x})$ can be written as $y\rho$ (where $\rho \geq 0$ and $y \in \{-1, +1\}$ is the class label of $\boldsymbol{x}$). In this case, the margin between the positive and negative margin borders becomes $\frac{2\rho}{\|\boldsymbol{w}\|_2}$. In the C-SVM, ρ is fixed as the value unity, and the margin specified in the C-SVM is controlled by the parameter, C. Alternatively, it can be controlled by adjusting the coefficient of ρ as in the ν-SVM proposed by the authors of [**?**]. The primal form of the optimization involved for the ν-SVM can be formulated as:

$$\begin{aligned} \min_{\boldsymbol{w},b,\rho,\boldsymbol{\xi}} & \frac{1}{2}\|\boldsymbol{w}\|_2^2 - C_0\nu\rho + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{\xi} \\ \text{s.t. } & \boldsymbol{Z}^{\mathrm{T}}\boldsymbol{w} + b\boldsymbol{y} \geq \rho\boldsymbol{1} - \boldsymbol{\xi} \\ & \boldsymbol{\xi} \geq 0 \\ & \rho \geq 0, \end{aligned} \tag{A.8}$$

where C_0 and ν are pre-specified parameters, and $\boldsymbol{C}$ is a column vector that takes instant value $C = \frac{C_0}{n}$ (we shall later show that C_0 can be simply set to unity later). $\boldsymbol{Z}$ is the sign-changed training set as in C-SVM.

As in the case of the C-SVM, by considering the Lagrangian function for this optimization and the KKT conditions, we can obtain the dual form of the optimization:

$$\begin{aligned} \min_{\boldsymbol{\mu}} & g(\boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{Z}^{\mathrm{T}}\boldsymbol{Z}\boldsymbol{\mu} \\ \text{s.t. } & \boldsymbol{y}^{\mathrm{T}}\boldsymbol{\mu} = 0 \\ & \boldsymbol{1}^{\mathrm{T}}\boldsymbol{\mu} \geq C_0\nu \\ & 0 \leq \boldsymbol{\mu} \leq \boldsymbol{C}. \end{aligned} \tag{A.9}$$

One can show that the optimal solution to $\boldsymbol{w}$ of the primal form as per the KKT condition: $\boldsymbol{w} = \boldsymbol{Z}_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}}$, where $\mathcal{S}$ is the set of indices of nonzero multipliers. If we first determine s positive points, denoted by $\boldsymbol{X}_+$, which are on the positive border, and s negative points, denoted by $\boldsymbol{X}_-$, which are on the negative border, we have:

$$\boldsymbol{X}_+^{\mathrm{T}}\boldsymbol{w} + b\boldsymbol{1} = \boldsymbol{\rho} = -\boldsymbol{X}_-^{\mathrm{T}}\boldsymbol{w} - b\boldsymbol{1}, \tag{A.10}$$

whence we can obtain the optimal bias as:

$$\begin{aligned} b &= -\frac{1}{2}\mathrm{mean}((\boldsymbol{X}_+ + \boldsymbol{X}_-)^{\mathrm{T}}\boldsymbol{w}) \\ &= -\frac{1}{2}\mathrm{mean}((\boldsymbol{X}_+ + \boldsymbol{X}_-)^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}(\boldsymbol{y}_{\mathcal{S}} * \boldsymbol{\mu}_{\mathcal{S}})). \end{aligned} \tag{A.11}$$

Consequently, the linear function in the decision function is

$$\begin{aligned} f(\boldsymbol{x}) &= \boldsymbol{w}^{\mathrm{T}}\boldsymbol{x} + b \\ &= \boldsymbol{x}^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}(\boldsymbol{y}_{\mathcal{S}} * \boldsymbol{\mu}_{\mathcal{S}}) - \frac{1}{2}\mathrm{mean}((\boldsymbol{X}_+ + \boldsymbol{X}_-)^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}(\boldsymbol{y}_{\mathcal{S}} * \boldsymbol{\mu}_{\mathcal{S}})). \end{aligned} \tag{A.12}$$

From the KKT conditions, the ν-SVM has the following properties:

1. From the dual form, we can see that the objective function is homogeneous. Thus, scaling the variable $\boldsymbol{\mu}$ would not change the decision function. Therefore, we can simply set $C_0 = 1$. The last two constraints are now $\boldsymbol{1}^{\mathrm{T}}\boldsymbol{\mu} \geq \nu$ and $\boldsymbol{0} \leq \boldsymbol{\mu} \leq \frac{\boldsymbol{1}}{\boldsymbol{n}}$.
2. An error is defined as the training sample that resides on the wrong side of its margin border. If $\rho > 0$, then ν is an upper bound on the fraction of errors, which means that $\nu \geq \frac{n_e}{n}$, where n_e is the number of errors.
3. A support vector is defined as the training samples corresponding to the non-zero multipliers which correspond to the active constraints $\rho\boldsymbol{1} - \boldsymbol{\xi}_{\mathcal{S}} - \boldsymbol{Z}_{\mathcal{S}}^{\mathrm{T}}\boldsymbol{w} - b\boldsymbol{y}_{\mathcal{S}} = 0$, implying that $\boldsymbol{Z}_{\mathcal{S}}^{\mathrm{T}}\boldsymbol{w} + b\boldsymbol{y}_{\mathcal{S}} \leq \rho\boldsymbol{1}$. Therefore, the SVs are a subset of the training samples that lie either on the correct margin border or at the wrong side of the correct margin border. If $\rho > 0$, then ν is a lower bound on the fraction of SVs, or in other words, $\nu \leq \frac{n_S}{n}$.
4. The range of ν in the ν-SVM is $(0, 1)$, while the range of C in the C-SVM is $(0, +\infty)$. Therefore, in practice, it is more convenient to use the ν-SVM rather than the C-SVM when it concerns model selection.
5. The conclusions that we reported with regard to the C-SVM concerning the relation between multiplier and the corresponding point positions apply to the ν-SVM as well.

Appendix B. One-Class SVMs

The one-class classification problem involves identifying outliers or novelties when we are merely given a limited number of training points. The one-class SVM is an implementation of Vapnik's principle stating that we need to avoid solving a more general problem than what is actually needed [**?**]. Instead of estimating the distribution of the data, the one-class SVM simply estimates the boundary of the distribution which captures the main mass of the data. By virtue of this, the one-class SVM is also referred to as the *Support Vector Domain Description* (SVDD) [**?**].

The border of the domain is defined by a non-negative linear combination of the outliers. The SVDD determines the "support" of a multivariate distribution, where the support means the set of SVs lying on the bound. Indeed, the various models differ in terms of the shapes of the border. While Tax and Duin treated this boundary as a hypersphere [**?**], Schölkopf *et al*. merely considered a hyperplane [**?**] representation. Although both appear quite different in their primal forms, they can be seen to be equivalent under weak conditions, which can be observed in dual form. As our NB schemes utilizes them, both of these methods are introduced and described in fair detail below.

Appendix B.1. Hypersphere-based One-Class SVM

The main idea of the hypersphere-based SVDD, proposed by Tax and Duin [**?**], is the following. The original data points are implicitly mapped to a higher-dimensional feature space, where a hypersphere is learned in such a way that its volume is as small as possible, while the core mass of the data is simultaneously kept as small as possible. An indicator function is also learned by which the data points inside are marked to be positive (core data points), and the data points outside are marked to be negative (i.e., as *outliers*).

This optimization problem associated with the hypersphere-based SVDD can be formulated, in its primal form, as follows:

$$\begin{aligned} &\min_{R,\boldsymbol{\xi},\nu} \boldsymbol{C}^{\mathrm{T}}\boldsymbol{\xi} + \nu R \\ &\text{s.t. } \|\phi(\boldsymbol{x}_i) - \boldsymbol{c}\|_2^2 \le R + \xi_i \\ &\quad \xi_i \ge 0 \\ &\quad R > 0, \end{aligned} \tag{B.1}$$

where $\boldsymbol{c}$ is the center of the hypersphere, R is its squared radius, ξ_i is a slack variable representing the error, and vector $\boldsymbol{C}$ is constant with $C_i = \frac{1}{n}$.

Working now with the dual of the optimization, we see that the dual has the form:

$$\begin{aligned} &\min_{\mu} \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\mu} - \frac{\nu}{2}\boldsymbol{k}^{\mathrm{T}}\boldsymbol{\mu} \\ &\text{s.t. } \mathbf{1}^{\mathrm{T}}\boldsymbol{\mu} = \nu \\ &\quad 0 \le \boldsymbol{\mu} \le \boldsymbol{C}, \end{aligned} \tag{B.2}$$

where $\boldsymbol{K} = \phi(\boldsymbol{X})^{\mathrm{T}}\phi(\boldsymbol{X})$, and $\boldsymbol{k} = \mathrm{diag}(\boldsymbol{K})$.

We now denote $\mathcal{S} = \{i|\mu_i > 0, i = 1, \cdots, n\}$ as the set of indices of nonzero multipliers that correspond to points that lie on or outside the border. From the KKT conditions, we know that the centroid of the hypersphere is a sparse non-negative linear combination of the training data points, that is $\boldsymbol{c} = \frac{1}{s}\phi(\boldsymbol{X})\boldsymbol{\mu} = \frac{1}{\nu}\phi(\boldsymbol{X})_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}}$. We define $\mathcal{B} = \{i|0 < \mu_i < C, i = 1, \cdots, n\}$ as the subset of indices of points on the hypersphere. Then, we can obtain R, the squared radius, as follows:

$$\begin{aligned} R &= \frac{1}{|\mathcal{B}|}\sum_{b\in\mathcal{B}} \|\phi(\boldsymbol{x})_b - \boldsymbol{c}\|_2^2 \\ &= \frac{1}{|\mathcal{B}|}\Big(\mathrm{trace}(\boldsymbol{K}_{\mathcal{B}}) - \frac{2}{\nu}\mathrm{sum}(\phi(\boldsymbol{X})_{\mathcal{B}}^{\mathrm{T}}\phi(\boldsymbol{X})_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}})\Big) + \frac{1}{\nu^2}\boldsymbol{\mu}_{\mathcal{S}}^{\mathrm{T}}\boldsymbol{K}_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}}. \end{aligned} \tag{B.3}$$

Consequently, the decision function is the following indicator function:

$$d(\boldsymbol{x}) = \mathrm{sign}[f(\boldsymbol{x})], \tag{B.4}$$

where $f(\boldsymbol{x})$ is defined as below:

$$\begin{aligned} f(\boldsymbol{x}) &= R - \|\phi(\boldsymbol{x}) - \boldsymbol{c}\|_2^2 \\ &= R - (\phi^{\mathrm{T}}(\boldsymbol{x})\phi(\boldsymbol{x}) - \frac{2}{\nu}\phi^{\mathrm{T}}(\boldsymbol{x})\phi(\boldsymbol{X})_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}} + \frac{1}{\nu^2}\boldsymbol{\mu}_{\mathcal{S}}^{\mathrm{T}}\boldsymbol{K}_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}}). \end{aligned} \tag{B.5}$$

From the KKT conditions, we have the following important properties:

1. If $\mu_i > 0$, the data point $\boldsymbol{x}_i$ resides on or outside of the hypersphere. Such an $\boldsymbol{x}_i$, which possesses a corresponding $\mu_i > 0$, is called a *support vector*. Observe that we only refer to the points whose corresponding multipliers are *nonzero* as support vectors. This is because, from point itemized below, we know that if $\boldsymbol{x}_i$ is on the hypersphere, it is possible that $\mu_i = 0$.
2. If $0 < \mu_i < C$, then the data point $\boldsymbol{x}_i$ is on the hypersphere. However, the reverse is not true. We can only affirm that if the data point $\boldsymbol{x}_i$ is on the hypersphere, then $0 \le \mu_i \le C$.
3. If $\boldsymbol{x}_i$ resides outside of the hypersphere, then $\mu_i = C$. In that case, $\boldsymbol{x}_i$ is called an *outlier*. However, from the above, we can see that the reverse is not true.
4. As in the case of the two-class ν-SVM, ν is a lower bound on the fraction of SVs, and an upper bound on the fraction of outliers. That is $\frac{n_e}{n} \le \nu \le \frac{n_S}{n}$.

Appendix B.2. Hyperplane Based One-Class SVM

Since a hyperplane is a less complex hypersurface, Schölkopf *et al.* proposed, rather, to work towards determining a one-class hyperplane rather than a hypersphere in the higher-dimensional feature space [**?**]. We know that a hyperplane is defined by the function $f(\boldsymbol{x}) = \boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) - b$, where $b \ge 0$. In this case, the indicator function $g(\boldsymbol{x}) = \mathrm{sign}[f(\boldsymbol{x})]$ takes the value +1 for a small region that captures most of the data, and

-1 elsewhere. Because the distance from the origin to the hyperplane is $\frac{-b}{\|\boldsymbol{w}\|_2}$, the task of minimizing the negative distance is equivalent to maximizing the absolute distance, which, in turn, is equivalent to maximizing the corresponding margin. The objective task is therefore to minimize $\frac{1}{2}\|\boldsymbol{w}\|_2^2 - b$, as well as the loss.

The corresponding optimization problem associated with the hyperplane-based SVDD can be formulated, in its primal form, as follows:

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \boldsymbol{C}^{\mathrm{T}}\boldsymbol{\xi} - \nu b \\ \text{s.t.} & \phi(\boldsymbol{X})^{\mathrm{T}}\boldsymbol{w} - b\mathbf{1} + \boldsymbol{\xi} \geq 0 \\ & \boldsymbol{\xi} \geq 0 \\ & b > 0, \end{aligned} \tag{B.6}$$

where $\boldsymbol{C}$ is a constant vector with elements equal to $\frac{1}{n}$.

If we now consider the dual of the optimization, we see that it has the form:

$$\begin{aligned} \min_{\boldsymbol{\mu}} & \frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{\mu} \\ \text{s.t.} & \ \mathbf{1}^{\mathrm{T}}\boldsymbol{\mu} = \nu \\ & 0 \leq \boldsymbol{\mu} \leq \boldsymbol{C}. \end{aligned} \tag{B.7}$$

This leads us to the conclusion that the decision function is:

$$\begin{aligned} f(\phi(\boldsymbol{x})) &= \text{sign}[\boldsymbol{w}^{\mathrm{T}}\phi(\boldsymbol{x}) - b] \\ &= \text{sign}[\boldsymbol{\mu}_{\mathcal{S}}^{\mathrm{T}}\phi(\boldsymbol{X}_{\mathcal{S}})^{\mathrm{T}}\phi(\boldsymbol{x}) - b], \end{aligned} \tag{B.8}$$

where $\mathcal{S}$ is the set of indices of nonzero multipliers. In order to compute b, we need to determine the data points on the boundary. If $0 < \mu_i < C$, it implies that the data point $\boldsymbol{x}_i$ is on the boundary and thus $f(\boldsymbol{x}_i) = 0$. We can, therefore, find a set $\mathcal{B}$ that includes some points satisfying $0 < \mu_i < C$, using which we can compute b as:

$$b = \text{mean}(\boldsymbol{X}_{\mathcal{B}}^{\mathrm{T}}\boldsymbol{X}_{\mathcal{S}}\boldsymbol{\mu}_{\mathcal{S}}). \tag{B.9}$$

The hyperplane based one-class SVM has the following important characteristics:

1. The relationship between the multiplier and the position in the hypersphere-based SVDD also applies to such a hyperplane-based SVDD.
2. As in the case of the two-class SVM, it can be proven that ν can lead to an upper bound of the fraction of outliers, and a lower bound of the fraction of the support vectors, i.e., $\frac{n_e}{n} \leq \nu \leq \frac{n_S}{n}$.
3. ν equals $\frac{n_e}{n}$ and $\frac{n_S}{n}$ asymptotically with probability 1.
4. From the dual forms of both the hypersphere and hyperplane formulations, we can see that the hypersphere formulation is equivalent to the hyperplane formulation in the case when we use a constant $K(\boldsymbol{x}, \boldsymbol{x})$, because, in this case, the linear term in the objective becomes constant.

References

[1] T. Ban and S. Abe. Implementing multi-class classifiers by one-class classification methods. In *IJCNN*, pages 327–332, Piscataway,NJ, July 2006. IEEE.

[2] K.P. Bennett and E.J. Bredensteiner. Duality and geometry in SVM classifiers. In *ICML*, pages 57–64, San Francisco, CA, 2000. IMLS, Morgan Kaufmann.

[3] J. C. Bezdek and L. I. Kuncheva. Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems*, 16(12):1445 – 1473, 2001.

[4] C.-C. Chang and C-J. Lin. Training ν-support vector classifiers: Theory and algorithms. *Neural Computation*, 13:2119–2147, 2001.

[5] C-.C. Chang and C-.J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[6] C. L. Chang. Finding prototypes for nearest neighbor classifiers. In *IEEE Transactions on Computing*, volume 23, pages 1179–1184, 1974.

[7] I. Czarnowski. Cluster-based instance selection for machine classification. *Knowledge and Information Systems*, 30(1):113–133, 2012.

[8] B. V. Dasarathy. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*. IEEE Computer Society Press, Los Alamitos, 1991.

[9] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

[10] P. A. Devijver and J. Kittler. On the edited nearest neighbor rule. In *Fifth International Conference on Pattern Recognition*, pages 72–80, December 1980.

[11] W. Duch. Similarity based methods: a general framework for Classification, Approximation and Association. *Control and Cybernetics*, 29(4):937–968, 2000.

[12] G. M. Foody. The significance of border training patterns in classification by a feedforward neural network using back propogation learning. *International Journal of Remote Sensing*, 20(18):3549–3562, 1999.

[13] A. Frank and A. Asuncion. UCI machine learning repository. Technical report, University of California, Irvine, California, 2010.

[14] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, San Diego, second edition, 1990.

[15] S. Garcia, J. Derrac, J. Ramon Cano, and F. Herrera. Prototype selection for nearest neighbor classification: Taxonomy and empirical study. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 34(3):417–435, 2012.

[16] G. W. Gates. The reduced nearest neighbor rule. In *IEEE Transactions on Information Theory*, volume 18, pages 431–433, 1972.

[17] P. E. Hart. The condensed nearest neighbor rule. In *IEEE Transactions on Information Theory*, volume 14, pages 515–516, 1968.

[18] C.-W. Hsu, C.-C. Chang, and C.-J. Lin. A practical guide to support vector classification. Technical report, Department of

Computer Science, National Taiwan University, Taipei,Taiwan, 2003.

[19] J.J. Hull. A database for handwritten text recognition research. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(5):550–554, 1994.

[20] A. K. Jain, R. P. W. Duin, and J. Mao. Statistical pattern recognition: A review. *TAMI*, 22(1):4 – 37, May 2000.

[21] T. Joachims. Making large-scale support vector machine learning practical. In B. Scholkopf, C.J.C. Burges, and A.J. Somla, editors, *Advances in Kernel Methods: Support Vector Learning*, chapter 11, pages 169–184. MIT, 1998.

[22] P.M. Kim and B. Tidor. Subsystem identification through dimensionality reduction of large-scale gene expression data. *Genome Research*, 13:1706–1718, 2003.

[23] T. Kohonen. *Self-Organizing Maps*. Springer, Berlin, 1995.

[24] D. Lee and J. Lee. Domain described spport vector classifier for multi-classification problems. *Pattern Recognition*, 40:41–51, 2007.

[25] K.C. Lee, J. Ho, and D. Kriegman. Acquiring linear subspaces for dace recognition under variable lighting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):684–698, 2005.

[26] G. Li, N. Japkowicz, T. J. Stocki, and R. K. Ungar. Full border identification for reduction of training sets. In *Canadian AI*, pages 203–215, 2008.

[27] Y. Li and A. Ngom. The regularized linear models and kernels toolbox in MATLAB. https://sites.google.com/site/rlmktool.

[28] Y. Li and A. Ngom. The non-negative matrix factorization toolbox for biological data mining. *BMC Source Code for Biology and Medicine*, 8(1):10, 2013. https://sites.google.com/site/nmftool.

[29] M. Maleki, M. Aziz, and L. Rueda. Analysis of relevant physicochemical properties in obligate and non-obligate protein-protein interactions. In *4th IEEE International Conference on Bioinformatics and Biomedicine Workshops*, pages 345–351. IEEE, 2011.

[30] D. Michie, D. Spiegelhalter, and C. Taylor. *Machine Learning, Neural and Statistical Classification*. Ellis Horwood, New York, 1994.

[31] T. Mitchell. *Machine Learning*. McGraw Hill, Ohio, 1997.

[32] I. Naseem, R. Togneri, and M. Bennamoun. Linear regression for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2106–2112, 2010.

[33] B. Ploj, R. Harb, and M. Zorman. Border pairs method constructive mlp learning classification algorithm. *Neurocomputing*, 126:180–187, 2012.

[34] G. L. Ritter, H. B. Woodruff, S. R. Lowry, and T. L. Isenhour. An algorithm for a selective nearest neighbor rule. In *IEEE Transactions on Information Theory*, volume 21, pages 665–669, 1975.

[35] B. Scholkopf, J.C. Platt, J. Shawe-Taylor, A.J. Smola, and B.C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

[36] B. Scholkopf, A.J. Smola, B.C. Williamson, and P.L. Bartlett. New support vector algorithm. *Neural Computation*, 12:1207–1245, 2000.

[37] D.M.J. Tax and R.P.W. Duin. Support vector domain description. *Pattern Recognition Letters*, 20:1191–1199, 1999.

[38] A. Thomas and B. J. Oommen. Optimal order statistics-based "anti-Bayesian" parametric pattern classification for the exponential family. 2012. (To be submitted).

[39] A. Thomas and B. J. Oommen. The fundamental theory of optimal "anti-Bayesian" parametric pattern classification using order statistics criteria. *Pattern Recognition*, 46:376–388, 2013.

[40] A. Thomas and B. J. Oommen. A novel border identification algorithm based on an "anti-Bayesian" paradigm. In *International Conference on Computer Analysis of Images and Patterns*, volume 8047 of *Lecture Notes in Computer Science*, pages 196–203. Springer Berlin / Heidelberg, 2013.

[41] A. Thomas and B. J. Oommen. Order statistics-based parametric classification for multi-dimensional distributions. 2013. Submitted for Publication.

[42] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Statistical Science*, 18(1):104–117, Feb. 2003.

[43] I. Tomek. Two modifcations of CNN. *IEEE Trans. Syst., Man and Cybern.*, SMC-6(6):769 – 772, Nov. 1976.

[44] I. Triguero, J. Derrac, S. Garcia, and F. Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Trans. on Systems, Man and Cybernetics - Part C: App. and Re.*, 42:86–100, 2012.

[45] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 2000.

[46] D. Wang, J. Wu, H. Zhang, K. Xu, and M. Lin. Torwards enhancing centroid classifier for text classification – a border instance approach. *Neurocomputing*, 101:299–308, 2013.

[47] Q. Xie, C.A. Laszlo, and R. K. Ward. Vector quantization techniques for nonparametric classifier design. *IEEE Trans. Pattern Anal. and Machine Intell.*, PAMI-15(12):1326 – 1330, Dec. 1993.

[48] H. Xiong, J. Wu, L. Liu, and M. Li. LSVDD: Rare class analysis based on local support vector data description. *Systems Engineering - Theory & Practice*, 32(8):1784–1792, 2012.