Generalizing Version Space Support Vector Machines for Non-Separable Data

E.N. Smirnov
MICC-IKAT
Maastricht University
Maastricht 6200 MD
The Netherlands
smirnov@micc.unimaas.nl

I.G. Sprinkhuizen-Kuyper
MICC-IKAT
Maastricht University
Maastricht 6200 MD
The Netherlands
kuyper@hmicc.unimaas.nl

N.I. Nikolaev
Goldsmiths College
University of London
London SE14 6NW
United Kingdom
n.nikolaev@gold.ac.uk

Abstract

Although version space support vector machines (VSSVMs) are a successful approach to reliable classification [6], they are restricted to separable data. This paper proposes generalized VSSVMs (GVSSVMs) applicable for separable and non-separable data. We show that GVSSVMs can outperform existing reliable-classification approaches.

1 Introduction

Machine-learning classifiers were applied to many classification tasks. Nevertheless, only few classifiers were used in critical applications. This is due to the difficulty to determine if a particular instance classification is reliable [6].

The two most prominent approaches to reliable classification are the Bayesian framework [4, 5] and the typicalness framework [8]. Although both frameworks are useful, the Bayesian framework can be misleading and the typicalness framework is classifier dependent.

To overcome these problems for the case of separable data we argued to use version spaces [4] as an approach to reliable classification in [6]. The key idea was to construct version spaces containing hypotheses of the target class or of its close approximations. In this way the unanimous-voting classification rule of version spaces does not misclassify instances; i.e., classifications become reliable.

To construct version spaces with the property above we proposed a volume-extension approach. The approach extends the volumes of version spaces s.t. instance misclassifications are blocked. The approach and version spaces were realized using support vector machines (SVMs) [7]. The combination was called version space support vector machines (VSSVMs) [6].

Although VSSVMs are able to outperform existing approaches to reliable classification, they are restricted to separable data. In this paper we introduce generalized VSSVMs (GVSSVMs) that can be applied for separable and non-separable training data. GVSSVMs are defined as version spaces [4] in hypothesis spaces of hyperplanes that are allowed to err on the training data depending on the data separability. We demonstrate that GVSSVMs can be adjusted to output only reliable classifications. Experiments show that GVSSVMs are capable of outperforming existing approaches to reliable classification.

This paper is organized as follows. The task of reliable classification is defined in section 2. Sections 3 and 4 present version spaces and the volume-extension approach. SVMs are given in section 5. Section 6 introduces GVSSVMs, followed by experiments and a comparison in sections 7 and 8. Section 9 concludes the paper.

2 Task of Reliable Classification

Consider training instances $x_i \in \mathbb{R}^n$ with labels $y_i \in \{-1, +1\}$ of a target class s.t. $x_i$ is in set $I^+$ ($I^-$) if $y_i = +1$ ($y_i = -1$). Given a space $H$ of hypotheses $h : \mathbb{R}^n \rightarrow \{-1, +1\}$, the task is to find $h \in H$ classifying correctly instances. If correct classification of $x \in \mathbb{R}^n$ is not possible, $x$ is unclassified (indicated by label 0).

3 Version Spaces

Version spaces are sets of consistent hypotheses [4].

Definition 1 Given a hypothesis space $H$ and training data $\langle I^+, I^- \rangle$, the version space $VS(I^+, I^-)$ equals

$$\{ h \in H \mid \text{cons}(h, \langle I^+, I^- \rangle) \},$$

where $\text{cons}$ is the consistency predicate defined as follows:

$$\text{cons}(h, \langle I^+, I^- \rangle) \leftrightarrow (\forall x_i \in I^+ \cup I^-)(y_i = h(x_i)).$$
The version-space classification rule is the unanimous voting. Given a non-empty version space $VS(I^+, I^-)$, an instance $x$ receives a classification $y \in Y \cup \{0\}$ as follows:

$$y = \begin{cases} 1 & (\forall h \in VS(I^+, I^-)) h(x) = +1, \\ -1 & (\forall h \in VS(I^+, I^-)) h(x) = -1, \\ 0 & \text{otherwise}. \end{cases}$$

**Definition 2** The volume $V(VS(I^+, I^-))$ of a version space $VS(I^+, I^-)$ is the set of all instances that are not classified by $VS(I^+, I^-)$.

The unanimous-voting rule is implemented if version spaces can be tested for collapse [3]. By theorem 1 if version space $VS(I^+, I^-)$ is nonempty, all hypotheses $h \in VS(I^+, I^-)$ assign class +1 to instance $x$ iff $VS(I^+, I^- \cup \{x\})$ is empty. All hypotheses $h \in VS(I^+, I^-)$ assign class -1 to x iff $VS(I^+ \cup \{x\}, I^-)$ is empty.

**Theorem 1** If $VS(I^+, I^-) \neq \emptyset$, then for each instance $x$:

$$(\forall h \in VS(I^+, I^-))(h(x) = +1 \leftrightarrow VS(I^+, I^- \cup \{x\}) = \emptyset),$$

$$(\forall h \in VS(I^+, I^-))(h(x) = -1 \leftrightarrow VS(I^+ \cup \{x\}, I^-) = \emptyset).$$

The problem to test version spaces for collapse is equivalent to the consistency problem [3]. The consistency problem is to determine the existence of a hypothesis $h \in H$ consistent with data. Hence, the unanimous-voting rule of version spaces can be implemented by any algorithm for the consistency problem. An algorithm for the consistency problem is called a consensus algorithm.

### 3.1 Analysis of Reliable Classification

Version spaces are sensitive w.r.t. class noise and expressiveness of the hypothesis space $H$ [4]. The class noise indicates that some class labels might be incorrect. The expressiveness of $H$ indicates if the hypothesis $h_t$ of the target class is in $H$.

Below we analyze the version-space classification.

**Case 1: Non-noisy Training Data and Expressive Hypothesis Space.** In this case $h_t \in VS(I^+, I^-)$. Thus, if instance $x$ is classified by $VS(I^+, I^-)$, $x$ is classified by $h_t$; i.e., $x$ is classified correctly (reliably).

**Case 2: Noisy Training Data.** In this case the set $I^+(I^+)$ is a union of a noise-free set $I^+_n$ ($I^+_t$) and a noisy set $I^+_n$ ($I^+_t$). The noisy data $(I^+_n, I^-_n)$ cause removal of version space $NVS = \{ h \in VS(I^+_t, I^-_t) | -cons(h, (I^+_n, I^-_n)) \}$ from $VS(I^+_t, I^-_t)$. Thus, $VS(I^+, I^-)$ classifies correctly (reliably) instances classified by $VS(I^+_t, I^-_t)$, but err on some instances in the volume of $NVS$.

**Case 3: Inexpressive Hypothesis Space.** Since $h_t \notin H$, some instances $x$ can be misclassified by all hypotheses in $VS(I^+, I^-)$.

### 4 Volume-Extension Approach

The volume-extension approach introduced in [6] is a new approach to overcome the problems with noisy training data and inexpressive hypothesis spaces. If a version space $VS(I^+, I^-) \subseteq H$ misclassifies instances, the approach is to find a new hypothesis space $H'$ s.t. the volume of version space $VS(I^+, I^-) \subseteq H'$ grows and blocks instance misclassifications. By theorem 2 to find $H'$ with such a property it is sufficient to guarantee that for all $\langle I^+, I^- \rangle$ if there is a consistent hypothesis $h \in H$, then there is a consistent hypothesis $h' \in H'$.

**Theorem 2** Consider hypothesis spaces $H$ and $H'$ s.t. for all $\langle I^+, I^- \rangle$ if there is $h \in H$ consistent with $(I^+, I^-)$, then there is $h' \in H'$ consistent with $(I^+, I^-)$ as well. Then, for all $(I^+, I^-)$ we have $V(VS(I^+, I^-)) \subseteq V(VS'(I^+, I^-))$.

The volume-extension approach is applied for cases 2-3.

**Case 2:** since the volume $V$ of $NVS$ is the error region for $VS(I^+, I^-)$, we have to search for $H'$ s.t. the volume of $VS'(I^+, I^-)$ comprises maximally $V$.

**Case 3:** since the cause of misclassifications is $h_t \notin H$, we have to search for $H'$ s.t. $VS'(I^+, I^-)$ includes more hypotheses approximating $h_t$. This means that if $x$ is misclassified, we define $H'$ s.t. $VS'(I^+, I^-)$ includes a hypothesis classifying $x$ as $h_t$ does. Thus, $x$ is not classified, so the misclassification is blocked.

We conclude that volume-extension approach can block misclassifications for cases 2-3.

### 5 Support Vector Machines

Support Vector Machines (SVMs) construct hyperplanes $h(p, C, (I^+, I^-))$ in the hypothesis space $H(p)$ of oriented hyperplanes s.t. the margin between training sets $I^+$ and $I^-$ is maximized given kernel parameter $p$ and parameter $C$ [7]. The parameter $C$ controls the trade-off between the margin and the training errors and penalizes the training errors. Thus, the probability that $h(p, C, (I^+, I^-))$ is consistent with $\langle I^+, I^- \rangle$ increases with $C$.

### 6 Generalized Version Space Support Vector Machines

This section introduces generalized version space support vector machines (GVSSVMs). Their hypothesis space, definition, and classification algorithm are presented.

**6.1 Hypothesis Space**

The version-space classification rule can be implemented by any consistency algorithm [3]. The key idea...
of GVSSVMs is to use a SVM as a consistency algorithm. Since a SVM is not a consistency algorithm in the hypothesis space $H(p)$ \cite{7}, we define a hypothesis space for which SVM is a consistency algorithm. Below we define this hypothesis space for separable and non-separable data.

### 6.1 Separable Data.

The training data $\langle I^+, I^- \rangle$ are separable, if the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+, I^- \rangle$. In this case by theorem 1 to classify an instance $x$ we need a consistency algorithm only for data sets $\langle I^+ \cup \{x\}, I^- \rangle$ and $\langle I^+, I^- \cup \{x\} \rangle$ for any $x$. Therefore, we define a hypothesis space $H(p, C, \langle I^+, I^- \rangle)$ for which SVM is a consistency algorithm for $\langle I^+ \cup \{x\}, I^- \rangle$ and $\langle I^+, I^- \cup \{x\} \rangle$ only.

**Definition 3** Given parameters $p$ and $C$ and data $\langle I^+, I^- \rangle$, if $\text{cons}(h(p,C,\langle I^+, I^- \rangle), \langle I^+, I^- \rangle)$, then $H(p, C, \langle I^+, I^- \rangle)$ equals:

$$\{ h \in H(p) | h = h(p, C, \langle I^+, I^- \rangle) \lor \exists x \in A(h = h(p, C, \langle I^+, I^- \rangle)) \land \text{cons}(h, \langle I^+ \cup \{x\}, I^- \rangle) \lor \exists x \in A(h = h(p, C, \langle I^+, I^- \cup \{x\} \rangle) \land \text{cons}(h, \langle I^+, I^- \cup \{x\} \rangle) \}$$

otherwise, $H(p, C, \langle I^+, I^- \rangle) = \emptyset$.

SVMs have an efficient consistency test for $H(p, C, \langle I^+, I^- \rangle)$ w.r.t. $\langle I^+ \cup \{x\}, I^- \rangle$ and $\langle I^+, I^- \cup \{x\} \rangle$ for any $x$. The test involves the hyperplanes $h(p, C, \langle I^+, I^- \rangle)$, $h(p, C, \langle I^+ \cup \{x\}, I^- \rangle)$, and $h(p, C, \langle I^+, I^- \cup \{x\} \rangle)$ only. It assumes that the instance-consistency property holds.

**Definition 4** SVM has the instance-consistency property w.r.t. data $\langle I^+, I^- \rangle$ if and only if for any instance $x$:

(i) if $h(p, C, \langle I^+ \cup \{x\}, I^- \rangle)$ is inconsistent with \( \langle I^+ \cup \{x\}, I^- \rangle \), then for all $x'$ hyperplanes $h(p, C, \langle I^+ \cup \{x'\}, I^- \rangle)$ and $h(p, C, \langle I^+, I^- \cup \{x'\} \rangle)$ are inconsistent with $\langle I^+ \cup \{x', I^- \rangle$;

(ii) if $h(p, C, \langle I^+ \cup \{x\}, I^- \rangle)$ is inconsistent with $\langle I^+, I^- \cup \{x\} \rangle$, then for all $x'$ hyperplanes $h(p, C, \langle I^+ \cup \{x'\}, I^- \rangle$ and $h(p, C, \langle I^+, I^- \cup \{x'\} \rangle$ are inconsistent with $\langle I^+ \cup \{x', I^- \rangle$.

We describe the SVM consistency test to decide if there are any hyperplane $h \in H(p, C, \langle I^+, I^- \rangle)$ consistent with $\langle I^+ \cup \{x\}, I^- \rangle$ for some $x$. For the test we first build the hyperplane $h(p, C, \langle I^+, I^- \rangle)$. If $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+ \cup \{x\}, I^- \rangle$, then there is an $h \in H(p, C, \langle I^+, I^- \rangle)$ consistent with $\langle I^+ \cup \{x\}, I^- \rangle$. If not, we check whether other hyperplanes in $H(p, C, \langle I^+, I^- \rangle)$ are consistent. We build the hyperplane $h(p, C, \langle I^+ \cup \{x\}, I^- \rangle)$. If $h(p, C, \langle I^+ \cup \{x\}, I^- \rangle)$ is consistent with $\langle I^+ \cup \{x\}, I^- \rangle$, then there is an $h \in H(p, C, \langle I^+, I^- \rangle)$ consistent with $\langle I^+ \cup \{x\}, I^- \rangle$. If not, by the instance-consistency property there is no $h \in H(p, C, \langle I^+, I^- \rangle)$ consistent with $\langle I^+ \cup \{x\}, I^- \rangle$.

The consistency test for hyperplanes in $H(p, C, \langle I^+, I^- \rangle)$ w.r.t. $\langle I^+ \cup \{x\}, I^- \rangle$ for any $x$. Below we formalize the SVM consistency test for $\langle I^+ \cup \{x\}, I^- \rangle$ and $\langle I^+, I^- \cup \{x\} \rangle$ for any $x$.

**Theorem 3** If the instance-consistency property holds and $H(p, C, \langle I^+, I^- \rangle) \neq \emptyset$, then for each instance $x$ we have:

$$\exists h \in H(p, C, \langle I^+, I^- \rangle)) \iff \text{cons}(h, \langle I^+ \cup \{x\}, I^- \rangle) \forall h(p, C, \langle I^+, I^- \rangle) \lor \text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+ \cup \{x\}, I^- \rangle) \lor \text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+, I^- \cup \{x\} \rangle)$$

### 6.1.2 Non-Separable Data.

The training data $\langle I^+, I^- \rangle$ are non-separable, if the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is not consistent with $\langle I^+, I^- \rangle$. In this case, by definition 3 $H(p, C, \langle I^+, I^- \rangle)$ is empty. Thus, we have to find a non-empty hypothesis space for which SVM is a consistency algorithm. Consider the corrected sets $I^+$ and $I^-$ induced from $I^+ \cup I^-$ by the decision boundary of $h(p, C, \langle I^+, I^- \rangle)$:

$$I^+ = \{ x \in I^+ \cup I^- | h(p, C, \langle I^+, I^- \rangle)(x) = +1 \}$$

$$I^- = \{ x \in I^+ \cup I^- | h(p, C, \langle I^+, I^- \rangle)(x) = -1 \}$$

In general, the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+ \cup I^- \rangle$. Thus, instead of the empty hypothesis space $H(p, C, \langle I^+, I^- \rangle)$, we use the non-empty hypothesis space $H(p, C, \langle I^+ \cup I^- \rangle)$ for which SVM is a consistency algorithm w.r.t. $\langle I^+ \cup \{x\}, I^- \rangle$ and $\langle I^+, I^- \cup \{x\} \rangle$ for any $x$.

We note that the same approach can be used for separable data. If the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+, I^- \rangle$, then $I^+ = I^+$ and $I^- = I^-$. And thus, $H(p, C, \langle I^+, I^- \rangle) \equiv H(p, C, \langle I^+, I^- \rangle)$. Therefore, we use $H(p, C, \langle I^+, I^- \rangle)$ when defining GVSSVMs for both separable and non-separable training data.

### 6.2 Definition

Given corrected data $\langle I^+, I^- \rangle$ induced by the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$, GVSSVMs are version spaces defined in $H(p, C, \langle I^+, I^- \rangle)$.
Input: An instance $x$ to be classified;
Training data sets $I^+$ and $I^-$;
The parameters $p$ and $C$ of SVM;

Output: classification of $x$:
Build a hyperplane $h(p, C; (I^+, I^-))$;
Form data $(I^+, I^-)$ using $h(p, C; (I^+, I^-))$;
if $\text{cons}(h(p, C; (I^+, I^-)), (I^+, I^-) \cup \{x\})$ then
Build hyperplane $h(p, C; (I^+, I^-) \cup \{x\})$;
if $\neg\text{cons}(h(p, C; (I^+, I^-) \cup \{x\}), (I^+, I^-) \cup \{x\})$
then return $+1$;
if $\text{cons}(h(p, C; (I^+, I^-)), (I^+, I^-) \cup \{x\})$
Build hyperplane $h(p, C; (I^+, I^-) \cup \{x\})$;
if $\neg\text{cons}(h(p, C; (I^+, I^-) \cup \{x\}), (I^+, I^-) \cup \{x\})$
then return $-1$;
return $0$.

Figure 1. The Classification Algorithm.

Definition 5 Consider a hypothesis space $H(p, C; \langle I^+, I^- \rangle)$ and training data $\langle I^+, I^- \rangle$ s.t. $I^+ \supseteq I^+$ and $I^- \supseteq I^-$. Then, the generalized version space support vector machine $\text{VSSVM}^p_C(I^+, I^-)$ is:

$$\{ h \in H(p, C; \langle I^+, I^- \rangle) | \text{cons}(h, \langle I^+, I^- \rangle) \}.$$

GVSSVMs are applied for separable and non-separable data. If the data are separable they equal VSSVMs.

6.3 Classification Algorithm

The classification algorithm implements the unanimous-voting rule on GVSSVMs. It assumes that the instance-consistency property holds, and, thus, GVSSVMs are tested for collapse using SVMs according to theorem 3.

The classification algorithm is given in figure 1. Assume that an instance $x$ is to be classified. First, the SVM hyperplane $h(p, C; (I^+, I^-))$ is built and used to form corrected data $\langle I^+, I^- \rangle$. Next, the algorithm tests whether $h(p, C; (I^+, I^-))$ is consistent with $\langle I^+ \cup \{x\}, I^- \rangle$. If so, then $\text{VSSVM}^p_C(I^+ \cup \{x\}, I^-)$ is not empty and the algorithm builds a SVM hyperplane $h(p, C; (I^+, I^-) \cup \{x\})$. If this hyperplane is inconsistent with $\langle I^+ \cup \{x\}, I^- \rangle$, then by theorem 3 $\text{VSSVM}^p_C(I^+ \cup \{x\}, I^-)$ is empty. Since $\text{VSSVM}^p_C(I^+ \cup \{x\}, I^-)$ is not empty and $\text{VSSVM}^p_C(I^+ \cup \{x\}, I^-)$ is empty, by theorem 1 the algorithm assigns class $+1$ to $x$. If class $+1$ cannot be assigned, the algorithm checks analogously if it can assign class $-1$. If no classification is assigned to $x$, label 0 is returned.

6.4 The Volume-Extension Approach

The volume-extension approach for GVSSVMs is applied for cases 2-3 using the parameter $C$ of SVMs.

Note that $\text{VSSVM}^p_C(I^+, I^-)$ also depends on $\langle I^+, I^- \rangle$.

Figure 2. Volume of GVSSVMs in $\mathbb{R}^2$ for $C = 30$ and $C = 1000$. Instances in $I^+$ and $I^-$ are marked by $\triangle$ and $\triangledown$, respectively. The volume of the GVSSVM for $C = 30$ is given by $\square$ boxes. The volume of the GVSSVM for $C = 1000$ is bounded by the lines.

We experimented with GVSSVMs using the polynomial kernel (the kernel parameter is the exponent $exp$). The evaluation method was 10-fold cross validation. We evaluated the accuracy rate $A$ and rejection rates $R$ of GVSSVMs (rejection rate is the proportion of unclassified instances). The $A/R$ graphs of GVSSVMs are shown in figure 3 for 6 binary UCI datasets [1]. One point in the graphs represents a GVSSVM for $exp = 1.5$ and some value of the parameter $C$. Subsequent points represent GVSSVMs for the same value of $exp$ and increased values of $C$ ($C \geq 1$). Hence, the graphs show the potential of GVSSVMs for reliable classification w.r.t. $C$; i.e., they show the evolution of the accuracy and rejection rates of GVSSVMs when the volume-extension approach is applied. From the graphs we may conclude that GVSSVMs are able to reach the desired 1.0 accuracy rate needed for reliable classification.
Figure 3. The accuracy/rejection graphs of GVSSVMs (solid line), NB (dotted line), and TNB (dashed line).

We note that GVSSVMs reject a fraction of instances even for small values of $C$ since they consist of more than one hyperplane (e.g., the labor data results in a minimal rejection rate of 0.7 and it is immediately classified reliably).

8 Comparison with Relevant Work

**Bayesian Framework.** The Bayesian framework [5] is the first approach used for reliable classification. This is due to the fact that the posterior class probabilities are natural estimates of the instance-classification reliability. These probabilities are formed from prior probabilities. Since it is difficult to estimate plausibly the prior probabilities, the Bayesian framework can be misleading [8].

**Typicalness Framework.** The typicalness framework [8] provides confidence values for each possible classification of instance $x_{l+1}$. The idea is to postulate a class $\hat{y} \in Y$ and to measure how likely it is that all elements in the extended sequence $\langle (x_1, y_1), \ldots, (x_l, y_l), (x_{l+1}, \hat{y}) \rangle$ are drawn from the same unknown distribution. The more typical the sequence is, the higher the confidence in $\hat{y}$.

We show the accuracy/rejection graphs of the Naive Bayes classifier (NB) and a typicalness classifier based on NB (TNB) in figure 3. The graphs are built by increasing thresholds on the posterior probabilities of NB and the confidence values of TNB using the 10-fold cross validation.

If the graphs of the GVSSVMs, NB, and TNB are compared we conclude that GVSSVMs outperform NB and TNB if the accuracy rate of 1.0 has to be reached.

9 Conclusion

We showed that GVSSVMs can provide reliable classifications and they can outperform existing approaches to reliable classification. The future research will focus on extending GVSSVMs for non-binary classification tasks and speeding up GVSSVMs using incremental SVMs [2].

References