

Version Space Support Vector Machines: An Extended Paper

E.N. Smirnov¹, I.G. Sprinkhuizen-Kuyper¹, G.I. Nalbantov², and S. Vanderlooy¹

Abstract. We argue to use version spaces as an approach to reliable classification. The key idea is to construct version spaces containing the hypotheses of the target concept or of its close approximations. As a result the unanimous-voting classification rule of version spaces does not misclassify; i.e., instance classifications become reliable.

We propose to implement version spaces using support vector machines. The resulting combination is called version space support vector machines (VSSVMs). Experiments show that VSSVMs are able to outperform the existing approaches to reliable classification.

1 Introduction

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

The two most prominent approaches to reliable classification are the Bayesian framework [7] and the typicalness framework [9] (see section 8). The Bayesian framework is a natural approach to reliable classification but it can be misleading if priors cannot be plausibly estimated [9]. The typicalness framework overcomes this problem but it depends heavily on the learning algorithm used.

To overcome these problems of the presented frameworks we argue to use version spaces [7] as an approach to reliable classification. The key idea is to construct version spaces containing hypotheses of the target concept to be learned or of its close approximations. In this way the unanimous-voting rule of version spaces does not misclassify instances; i.e., instance classifications become reliable.

We analyze the instance classification of version spaces for the case when data is non-noisy and hypothesis space is expressive as well as for the opposite three cases. For the latter instance classification can be unreliable and we propose a volume-extension approach. The approach is to grow the volumes of version spaces s.t. instance misclassifications are blocked.

We propose implementing version spaces for reliable classification using support vector machines (SVMs) [8]. Their combination is called version space support vector machines (VSSVMs). We apply the volume-extension approach on VSSVMs. In experiments VSSVMs outperform the existing reliable-classification approaches.

This paper is as follows. The task of reliable classification is defined in section 2. Section 3 considers version spaces for reliable classification and the volume-extension approach. SVMs are described in section 4. Section 5 introduces VSSVMs. The volume-extension

approach for VSSVMs is in section 6. Section 7 and 8 present experiments and a comparison. Section 9 concludes the paper.

2 Task of Reliable Classification

Consider l different training instances \mathbf{x}_i in \mathbb{R}^n . Each \mathbf{x}_i has a class label $y_i \in Y$ w.r.t. a binary target concept, i.e., $Y = \{-1, +1\}$. The labels separate the instances into two sets I^+ and I^- ($\mathbf{x}_i \in I^+$ iff $y_i = +1$; $\mathbf{x}_i \in I^-$ iff $y_i = -1$). Given a space H of hypotheses h ($h : \mathbb{R}^n \rightarrow Y$), the task of reliable classification is to find $h \in H$ that correctly classifies future, unseen instances. When correct classification is not possible, the classification process outputs 0.

3 Version Spaces

Version spaces are sets of hypotheses consistent with data [7].

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

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

where $\text{cons}(h, \langle I^+, I^- \rangle) \leftrightarrow (\forall \mathbf{x}_i \in I^+ \cup I^-) y_i = h(\mathbf{x}_i)$.

The version-space classification rule is the unanimous voting. Given a version space $VS(I^+, I^-)$, an instance \mathbf{x} receives a classification $y \in Y \cup \{0\}$ as follows:

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

Definition 2 Volume $V(VS(I^+, I^-))$ of 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 [4]. By theorem 1 if version space $VS(I^+, I^-)$ is nonempty, all hypotheses $h \in VS(I^+, I^-)$ assign class +1 to instance \mathbf{x} iff $VS(I^+, I^- \cup \{\mathbf{x}\})$ is empty. All $h \in VS(I^+, I^-)$ assign class -1 to \mathbf{x} iff $VS(I^+ \cup \{\mathbf{x}\}, I^-)$ is empty.

Theorem 1 If $VS(I^+, I^-) \neq \emptyset$, then for each instance \mathbf{x} :
 $(\forall h \in VS(I^+, I^-)) h(\mathbf{x}) = +1 \leftrightarrow VS(I^+, I^- \cup \{\mathbf{x}\}) = \emptyset$,

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

¹ MICC-IKAT, Universiteit Maastricht, Maastricht 6200 MD, The Netherlands, email: {smirnov, kuyper, s.vanderlooy}@cs.unimaas.nl

² ERIM, Erasmus University Rotterdam, Rotterdam 3000 DR, The Netherlands, email: nalbantov@few.eur.nl

The problem to test version spaces for collapse is equivalent to the consistency problem [4]. 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 consistency algorithm.

3.1 Analysis of Reliable Classification

Version spaces are sensitive w.r.t. class noise in training data and expressiveness of hypothesis space H [7]. Class noise indicates that the class labels of some instances are incorrect. Expressiveness of the space H indicates if the hypothesis h_t of the target concept is in H .

Below we analyze instance classification with version spaces.

Case 1: Non-noisy Training Data and Expressive Hypothesis Space. Since H is expressive, $h_t \in H$. Since the training data $\langle I^+, I^- \rangle$ are non-noisy, h_t is consistent with $\langle I^+, I^- \rangle$. Thus, by definition 1 $h_t \in VS(I^+, I^-)$. In this way, if an instance \mathbf{x} is classified by $VS(I^+, I^-)$, \mathbf{x} is classified by h_t ; i.e., \mathbf{x} is classified correctly. Thus, version spaces output only reliable classifications.

Case 2: Noisy Training Data. If there is noise, the set I^+ (I^-) is a union of a noise-free set I_f^+ (I_f^-) and a noisy set I_n^+ (I_n^-). The noisy data $\langle I_n^+, I_n^- \rangle$ cause removal of version space $NVS = \{h \in VS(I_f^+, I_f^-) \mid \neg \text{cons}(h, \langle I_n^+, I_n^- \rangle)\}$ from $VS(I_f^+, I_f^-)$. Thus, the resulting $VS(I^+, I^-)$ classifies correctly instances classified by $VS(I_f^+, I_f^-)$, but it errs on some instances in the volume of NVS .

Case 3: Inexpressive Hypothesis Space. If the hypothesis space H is inexpressive ($h_t \notin H$), it is possible that the hypotheses in $VS(I^+, I^-)$ do not approximate the target concept well; i.e., there may exist an instance \mathbf{x} that is misclassified by all hypotheses in $VS(I^+, I^-)$. Thus, $VS(I^+, I^-)$ may misclassify instances.

Case 4: Noisy Training Data and Inexpressive Hypothesis Space. This case is a union of cases 2 and 3.

3.2 Volume-Extension Approach

The volume-extension approach 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 $\langle I^+, I^- \rangle$, then there is $h' \in H'$ consistent with $\langle I^+, I^- \rangle$ as well. Then, for all $\langle I^+, I^- \rangle$ we have $V(VS(I^+, I^-)) \subseteq V(VS'(I^+, I^-))$.

Below we analyze the volume-extension approach for cases 2-4.

Case 2: since the volume 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 the volume of NVS ;

Case 3: since the causes of misclassification for $VS(I^+, I^-)$ are the hypotheses in $VS(I^+, I^-)$ not approximating the target concept well, we have to search for H' s.t. $VS'(I^+, I^-)$ includes more hypotheses approximating the target concept better. This means that if we have an instance \mathbf{x} misclassified by $VS(I^+, I^-)$, we define H' s.t. $VS'(I^+, I^-)$ includes a hypothesis classifying \mathbf{x} as the target concept. Thus, \mathbf{x} is not classified, so the misclassification is blocked.

Case 4: the explanations for cases 2 and 3 hold here.

We conclude that the volume-extension approach can block instance misclassification for cases 2-4. This result implies that *version spaces can be viewed as an approach to reliable classification*.

4 Support Vector Machines

Support Vector Machines (SVMs) were proposed for classification tasks [8]. SVM constructs a hyperplane used as a decision boundary for classification. The parameters of the SVM hyperplane are derived from the solution of the following optimization problem:

$$\max_{\alpha} \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \quad (1)$$

$$\text{subject to } 0 \leq \alpha_i \leq C, i = 1, 2, \dots, l, \text{ and } \sum_{i=1}^l y_i \alpha_i = 0,$$

where $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)' \phi(\mathbf{x}_j)$ is a kernel function that calculates inner products of instances \mathbf{x}_i and \mathbf{x}_j in a higher dimensional feature space \mathbb{F} and ϕ is a mapping from \mathbb{R}^n to \mathbb{F} . Maximizing the term $-\sum_{i,j=1}^l \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j)$ corresponds to maximizing the margin between the two classes. The parameter C determines the trade-off between the margin and the amount of training errors. The alphas are the weights associated with the training instances. All instances with nonzero weights are ‘‘support vectors’’. They determine the SVM hyperplane consisting of all points \mathbf{x} which satisfy $\sum_{i=1}^l y_i \alpha_i k(\mathbf{x}_i, \mathbf{x}) + b = 0$. The parameter b is found from the KKT conditions associated with (1).

The hypothesis space of SVMs is the set of all oriented hyperplanes in \mathbb{R}^n or in a higher dimensional feature space \mathbb{F} . The hypothesis space of SVMs is denoted by $H(p)$ where p is a kernel parameter. For the RBF kernel p is gamma and for the polynomial kernel p is the exponent. The SVM hyperplane is denoted by $h(p, C, \langle I^+, I^- \rangle)$.

We consider the asymptotic behaviors of SVMs w.r.t. the parameter C [5]. When C increases, the weight of training errors increases, while other things stay equal. Consequently, the SVM algorithm will try to find a new balance between the margin width and amount of training errors. In particular, the margin will decrease and the amount of classification errors will generally go down. Therefore, for all data $\langle I^+, I^- \rangle$ the probability that $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+, I^- \rangle$ increases with the parameter C .

5 Version Space Support Vector Machines

This section introduces version space support vector machines (VSSVMs). Their hypothesis space, definition, and classification algorithm are given in subsections 5.1, 5.2, and 5.3, respectively.

5.1 Hypothesis Space

The version-space classification rule can be realized by any consistency algorithm [4]. The key idea of VSSVMs is to use a SVM as a consistency algorithm. Theorem 1 shows that for training data $\langle I^+, I^- \rangle$ we need a consistency algorithm only for data sets $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ for any \mathbf{x} . Since a SVM is not a consistency algorithm in the hypothesis space $H(p)$ [8], below we define a hypothesis space $H(p, C, \langle I^+, I^- \rangle)$ for which SVM is a consistency algorithm w.r.t. $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ for any \mathbf{x} .

$H(p, C, \langle I^+, I^- \rangle)$ is defined if the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+, I^- \rangle$. It includes $h(p, C, \langle I^+, I^- \rangle)$ and all the SVM hyperplanes $h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^- \rangle)$ and $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$ for any \mathbf{x} that are consistent with their training data.

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:

$$\{h \in H(p) \mid h = h(p, C, \langle I^+, I^- \rangle) \vee (\exists \mathbf{x})(h = h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^-)) \wedge \text{cons}(h, \langle I^+ \cup \{\mathbf{x}\}, I^-)) \vee (\exists \mathbf{x})(h = h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle) \wedge \text{cons}(h, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle))\},$$

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. $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ for any \mathbf{x} . The test involves the hyperplanes $h(p, C, \langle I^+, I^- \rangle)$, $h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^- \rangle)$, and $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$ 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 \mathbf{x} :

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

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

The consistency test for hyperplanes in $H(p, C, \langle I^+, I^- \rangle)$ w.r.t. $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ for any \mathbf{x} is analogous. Thus, SVM is a consistency algorithm in $H(p, C, \langle I^+, I^- \rangle)$ w.r.t. $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ for any \mathbf{x} . Below we formalize the SVM consistency test for $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ for any \mathbf{x} in theorem 3.

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

$$\begin{aligned} (\exists h \in H(p, C, \langle I^+, I^- \rangle)) \text{cons}(h, \langle I^+ \cup \{\mathbf{x}\}, I^- \rangle) &\leftrightarrow \\ &[\text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+ \cup \{\mathbf{x}\}, I^-) \vee \\ &\text{cons}(h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^- \rangle), \langle I^+ \cup \{\mathbf{x}\}, I^-)], \\ (\exists h \in H(p, C, \langle I^+, I^- \rangle)) \text{cons}(h, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle) &\leftrightarrow \\ &[\text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+, I^- \cup \{\mathbf{x}\} \rangle) \vee \\ &\text{cons}(h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle), \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)]. \end{aligned}$$

By theorem 3 to test if there is a hyperplane $h \in H(p, C, \langle I^+, I^- \rangle)$ consistent with $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$ test if either of hyperplanes $h(p, C, \langle I^+, I^- \rangle)$ and $h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^- \rangle)$ is consistent with $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$. Testing if there is a hyperplane $h \in H(p, C, \langle I^+, I^- \rangle)$ consistent with $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$ is analogous.

5.2 Definition of VSSVMs

VSSVMs are version spaces defined in $H(p, C, \langle I^+, I^- \rangle)$.

Input: An instance \mathbf{x} to be classified;
 Training data sets I^+ and I^- ;
 The parameters p and C of SVM;
Output: classification of \mathbf{x} ;
 Build a hyperplane $h(p, C, \langle I^+, I^- \rangle)$;
if $\neg \text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+, I^- \rangle)$
then return 0;
if $\text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+ \cup \{\mathbf{x}\}, I^-)$ **then**
 Build hyperplane $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$;
if $\neg \text{cons}(h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle), \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$
then return +1;
if $\text{cons}(h(p, C, \langle I^+, I^- \rangle), \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$ **then**
 Build hyperplane $h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^-)$;
if $\neg \text{cons}(h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^-), \langle I^+ \cup \{\mathbf{x}\}, I^-)$
then return -1;
return 0.

Figure 1. The Classification Algorithm of VSSVMs.

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 version space support vector machine $VS_C^p(I^+, I^-)$ ³ is:

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

Since VSSVMs are version spaces, the inductive bias of VSSVMs is the restriction bias [7]. Since parameters p and C as well as training data define the hypothesis space $H(p, C, \langle I^+, I^- \rangle)$, they control the inductive bias of VSSVMs.

5.3 Classification Algorithm

The classification algorithm of VSSVMs implements the unanimous-voting rule and is based on theorem 1. It assumes that the instance-consistency property holds. Thus, to test version spaces for collapse SVMs are employed according to theorem 3.

The classification algorithm is given in figure 1. Assume that an instance \mathbf{x} is to be classified. Then, the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is built. If $h(p, C, \langle I^+, I^- \rangle)$ is inconsistent with $\langle I^+, I^- \rangle$, according to definition 3 $H(p, C, \langle I^+, I^- \rangle) = \emptyset$; i.e., $VS_C^p(I^+, I^-) = \emptyset$. Thus, according to the unanimous-voting rule the algorithm returns 0; i.e., the classification of \mathbf{x} is unknown. If the hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+, I^- \rangle$, then $VS_C^p(I^+, I^-) \neq \emptyset$. In this case the algorithm tests whether $h(p, C, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+ \cup \{\mathbf{x}\}, I^- \rangle$. If so, then $VS_C^p(I^+ \cup \{\mathbf{x}\}, I^-) \neq \emptyset$ and the algorithm builds a SVM hyperplane $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$. If $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\} \rangle)$ is inconsistent with $\langle I^+, I^- \cup \{\mathbf{x}\} \rangle$, then by theorem 3 $VS_C^p(I^+, I^- \cup \{\mathbf{x}\}) = \emptyset$. Since $VS_C^p(I^+ \cup \{\mathbf{x}\}, I^-) \neq \emptyset$ and $VS_C^p(I^+, I^- \cup \{\mathbf{x}\}) = \emptyset$, by theorem 1 the algorithm assigns class +1 to \mathbf{x} . If class +1 cannot be assigned, the algorithm checks analogously if it can assign class -1. If no classification is assigned to \mathbf{x} , 0 is returned.

6 The Volume-Extension Approach for VSSVMs

To overcome the problems of noisy training data and inexpressive hypothesis spaces for VSSVMs we propose applying our

³ Note that $VS_C^p(I^+, I^-)$ also depends on $\langle I^+, I^- \rangle$.

volume-extension approach using the parameter C of SVMs. In section 4 we showed that the probability that the SVM hyperplane $h(p, C, \langle I^+, I^- \rangle)$ is consistent with data $\langle I^+, I^- \rangle$ increases with C . Thus, for two values C_1 and C_2 of parameter C s.t. $C_1 < C_2$ and arbitrary $\langle I^+, I^- \rangle$ the probability that $h(p, C_2, \langle I^+, I^- \rangle)$ is consistent with $\langle I^+, I^- \rangle$ is higher than that of $h(p, C_1, \langle I^+, I^- \rangle)$. This implies by theorem 2 that the probability of $V(VS_{C_1}^p(I^+, I^-)) \subseteq V(VS_{C_2}^p(I^+, I^-))$ increases. This means that the volume of VSSVMs increases with the parameter C (see figure 2).

Given training data $\langle I^+, I^- \rangle$ and kernel parameter p , applying the volume-extension approach means to find C for which $V(S_C^p(I^+, I^-)) \subseteq H(p, C, \langle I^+, I^- \rangle)$ classifies instances reliably. Since the volume of VSSVMs increases with C , we can find a minimal value for C in an internal validation process using binary search s.t. instances are classified reliably and the volume of $V(S_C^p(I^+, I^-))$ is minimized.

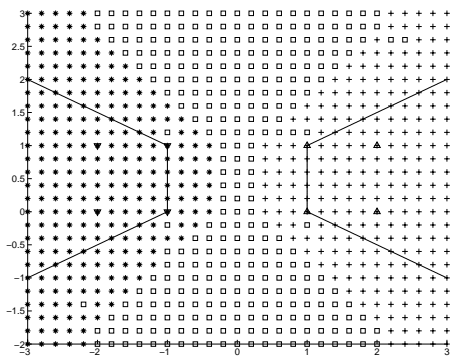


Figure 2. The volume of VSSVMs in \mathbb{R}^2 for $C = 30$ and $C = 1000$. Instances in I^+ are marked by \triangle , instances in I^- are marked by ∇ . The volume of the VSSVM for $C = 30$ is presented by \square boxes. The volume of the VSSVM for $C = 1000$ is bounded by the lines.

7 Experiments

We experimented with VSSVMs using the polynomial and RBF kernels. The evaluation method was the leave-one-out method. We evaluated four statistics of the classification performance of VSSVMs:

- positive and negative rejection rates PRr and NRr : proportion of positive and negative instances *unclassified* by VSSVMs;
- true positive and true negative rates TPr and TNr [3] on instances *classified* by VSSVMs.

The (TPr, PRr) and (TNr, NRr) graphs of VSSVMs are shown in the first and third columns of figure 3 for 7 binary UCI datasets [1]. One point in the graphs represents a VSSVM for some values of the parameters p and C . Subsequent points represent VSSVMs for the same value of p and increased values of C . Hence, the graphs show the potential of VSSVMs for reliable classification w.r.t. C .

The initial value p_0 of the kernel parameter p used in VSSVMs was chosen using sequential search s.t. p_0 is minimized and all $V(S_{C_0}^{p_0}(I^+ \setminus \{\mathbf{x}\}, I^-))$ and $V(S_{C_0}^{p_0}(I^+, I^- \setminus \{\mathbf{x}\}))$ ⁴ are nonempty

⁴ These are VSSVMs if an instance \mathbf{x} is left out in the leave-one-out validation.

for some value of the parameter C . The initial value C_0 of C was chosen using binary search given p_0 s.t. C_0 is minimized and all $V(S_{C_0}^{p_0}(I^+ \setminus \{\mathbf{x}\}, I^-))$ and $V(S_{C_0}^{p_0}(I^+, I^- \setminus \{\mathbf{x}\}))$ are nonempty.

The values p_0 and C_0 define VSSVMs represented as the most left points of the graphs on figure 3. The PRr_0 and NRr_0 of these VSSVMs are nonzero and the graphs are undefined in the intervals $[0, PRr_0)$ and $[0, NRr_0)$. Most initial VSSVMs have TPr_0 and TNr_0 lower than 1. This is due to noise in the datasets used in the experiments and/or inexpressive hypothesis spaces. To overcome these problems we applied our volume-extension approach by stepwise increasing the parameter C from C_0 to 10000. For each step we generated a VSSVM and plotted its (TPr, PRr) and (TNr, NRr) points on the graphs from figure 3. The graphs show that VSSVMs reach TPr and TNr of 1. The only exception is the VSSVM with the RBF kernel for the hepatitis dataset. Thus, we may conclude that the volume-extension approach is capable of solving the problems with noisy training data and inexpressive hypothesis spaces.

We compare VSSVMs for the polynomial kernel (VSSVM-P) and for the RBF kernel (VSSVM-RBF) w.r.t. reliable classification using the graphs of figure 3. For that purpose we use the minimal rejection rates PRr_m and NRr_m ⁵ for which TPr and TNr are 1. The PRr_m of VSSVM-P is lower than that of VSSVM-RBF for 5 out of 7 datasets. The NRr_m of VSSVM-P is lower than that of VSSVM-RBF for 2 out of 3 datasets. For the remaining 4 datasets NRr_m of VSSVM-P and VSSVM-RBF are equal. Thus, VSSVM-P is better for reliable classification than VSSVM-RBF in our experiments.

8 Comparison with Relevant Work

Bayesian Framework. The Bayesian framework [7] is the first approach used for reliable classification. This is due to the fact that the posterior class probabilities are natural estimates of the reliability of instance classification. These probabilities are computed from prior probabilities. Since it is difficult to estimate plausibly the prior probabilities, the Bayesian framework can be misleading [9].

Typicalness Framework. The typicalness framework [9] provides confidence values for each possible classification of an instance \mathbf{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 $\langle (\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), (\mathbf{x}_{l+1}, \hat{y}) \rangle$ are drawn from the same unknown distribution. The more typical the sequence is, the higher the confidence in \hat{y} .

In the second and fourth columns of figure 3 we present (TPr, PRr) and (TNr, NRr) graphs of the Naive Bayes classifier (NB) and a typicalness algorithm based on NB (Typ-NB). The graphs are constructed by stepwise increasing thresholds on the posterior probabilities of NB and the typicalness of Typ-NB using the leave-one-out method.

We compare VSSVMs, NB, and Typ-NB w.r.t. reliable classification (see figure 3). We use the minimal rejection rates PRr_m and NRr_m for which TPr and TNr are 1. The comparison shows that:

- (a) the PRr_m of VSSVM-P is lower than that of NB and Typ-NB for 4 out of 7 datasets and 5 out of 7 datasets, respectively. The NRr_m of VSSVM-P is lower than NRr_m of NB and Typ-NB for 5 out of 7 datasets. For the minority class the minimal rejection rates of VSSVM-P are lower than those of NB and Typ-NB for 5 out of 7 datasets and 6 out of 7 datasets, respectively. For the majority class the minimal rejection rates of VSSVM-P are lower than those of NB and Typ-NB for 4 out of 7 datasets.

⁵ If a classifier has lower values of PRr_m and NRr_m , it can classify more.

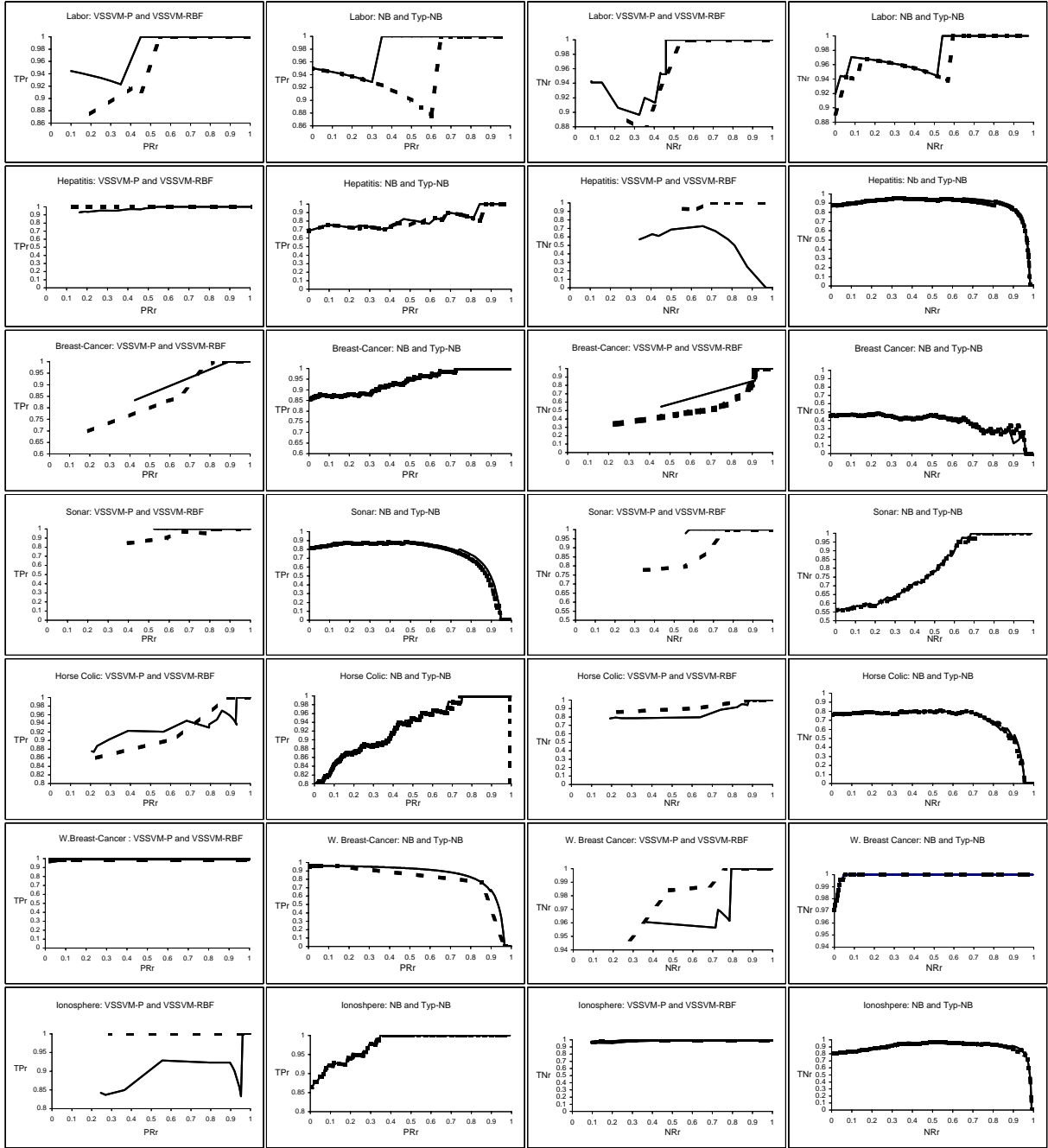


Figure 3. The (TPr, PRr) and (TNr, NRr) graphs of VSSVM-RBF (—), VSSVM-P (---), NB (—), and Typ-NB (---).

(b) the PRr_m of VSSVM-RBF is lower than that of NB and Typ-NB for 3 out of 7 datasets and 5 out of 7 datasets, respectively. The NRr_m of VSSVM-RBF is lower than that of NB and Typ-NB for 4 out of 6 datasets and 5 out of 6 datasets, respectively. For the minority class the minimal rejection rates of VSSVM-RBF are lower than those of NB and Typ-NB for 4 out of 7 datasets and 5 out of 7 datasets, respectively. For the majority class the minimal rejection rates of VSSVM-RBF are lower than those of NB and Typ-NB for 3 out of 6 datasets and 5 out of 6 datasets, respectively. The comparison is for 6 datasets for some cases since

for the hepatitis data VSSVM-RBF, NB, and Typ-NB do not have NRr_m .

From (a) and (b) we conclude that VSSVMs outperform NB and Typ-NB for the task of reliable classification in our experiments.

9 Conclusion

In this paper we showed that VSSVMs are able to provide reliable classifications when training data is noisy and hypothesis spaces are

inexpressive. This is due to the nature of VSSVMs and the volume-extension approach. The experiments show that VSSVMs are able to outperform the existing approaches to reliable classification.

We foresee three future research directions. The first one is to extend VSSVMs for non-binary classification tasks. The second direction is to extend VSSVMs for tasks for which no consistent hypotheses exist. The third direction is to speed up VSSVMs using incremental SVMs [2].

REFERENCES

- [1] C. Blake and C. Merz. UCI repository of ML databases, 1998.
- [2] G. Cauwenberghs and T. Poggio, 'Incremental support vector machine learning', in *Proceedings of NIPS*2000*, pp. 409–415, (2000).
- [3] Tom Fawcett, 'Roc graphs: Notes and practical considerations for researchers', Technical Report HPL-2003-4, HP Laboratories, (2003).
- [4] H. Hirsh, N. Mishra, and L. Pitt, 'Version spaces and the consistency problem', *Artificial Intelligence*, **156**(2), 115–138, (2004).
- [5] S. Keerthi and C. Lin, 'Asymptotic behaviors of support vector machines with gaussian kernel', *Neural Computation*, **15**, 1667–1689, (2003).
- [6] M. Kukar and C. Groelj, 'Transductive machine learning for reliable medical diagnostics', *J. med. syst.*, **29**(1), 13–23, (2005).
- [7] T.M. Mitchell, *Machine learning*, McGraw-Hill, New York, NY, 1997.
- [8] V. Vapnik, *Statistical Learning Theory*, John Wiley, NY, 1998.
- [9] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*, Springer, New York, NY, 2005.