Version Space Support Vector Machines

E.N. Smirnov[1], **I.G. Sprinkhuizen-Kuyper**[1], **G.I. Nalbantov**[2], and **S. Vanderlooy**[1]

1 Introduction

The task of reliable classification is to determine if a particular instance classification is reliable. There exist two approaches to the task: the Bayesian framework [3] and the typicalness framework [5]. Although both frameworks are useful, the Bayesian framework can be misleading and the typicalness framework is classifier dependent.

To overcome these problems we argue to use version spaces (VSs) [3] for reliable classification. The key idea is to construct VSs containing hypotheses of the target class or of its close approximations. In this way the unanimous-voting classification rule of VSs does not misclassify instances; i.e., instance classifications become reliable.

To construct VSs with the property above we propose an approach extending the volumes of VSs s.t. instance misclassifications are blocked. The approach and VSs are realized using support vector machines (SVMs) [4]. The combination is called version space support vector machines (VSSVMs). We show that VSSVMs are able to outperform the existing approaches to reliable classification.

2 Task of Reliable Classification

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

3 Version Spaces

Given a hypothesis space H and training data $\langle I^+, I^- \rangle$, version space $VS(I^+, I^-)$ is the set of hypotheses consistent with $\langle I^+, I^- \rangle$:

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

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

The unanimous-voting rule of VSs classifies instance $\mathbf{x}$ as follows:

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

The volume $V(VS(I^+, I^-))$ of $VS(I^+, I^-)$ is the set of instances not classified by the unanimous-voting rule ($y = 0$). By theorem 1 the rule can be implemented by testing VSs for collapse [2].

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

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

Testing VSs for collapse is the consistency problem [2]. The problem is to check the existence of a hypothesis $h \in H$ consistent with data. Hence, the VSs unanimous-voting rule can be implemented by any algorithm for the consistency problem (consistency algorithm).

4 Volume-Extension Approach

VSs are sensitive to the class noise in the training data and the expressiveness of hypothesis space H [3]. The expressiveness of H indicates if the hypothesis h_t of the target class is in H. We analyze VS instance classification with these two factors.

Case 1: no class noise and expressive hypothesis space. In this case $h_t \in VS(I^+, I^-)$. Thus, if instance $\mathbf{x}$ is classified by $VS(I^+, I^-)$, $\mathbf{x}$ is classified by h_t; i.e., $\mathbf{x}$ is classified correctly (reliably).

Case 2: class noise. In this case 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* $= \{h \in VS(I_f^+, I_f^-) | \neg cons(h, \langle I_n^+, I_n^- \rangle)\}$ from $VS(I_f^+, I_f^-)$. Thus, $VS(I^+, I^-)$ classifies correctly (reliably) instances classified by $VS(I_f^+, I_f^-)$, but errs on some instances in the volume of *NVS*.

Case 3: inexpressive hypothesis space. Since $h_t \notin H$, some instances $\mathbf{x}$ can be misclassified by all hypotheses in $VS(I^+, I^-)$.

The volume-extension approach is a new approach for cases 2-3. If $VS(I^+, I^-) \subseteq H$ misclassifies, then the approach is to find a new hypothesis space H' s.t. the volume of $VS'(I^+, I^-) \subseteq H'$ grows and blocks misclassifications. It is applied using theorem 2.

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 $\langle I^+, I^- \rangle$. Then, for all $\langle I^+, I^- \rangle$ the volume $V(VS(I^+, I^-))$ is a subset of the volume $V(VS'(I^+, I^-))$.*

Below we apply the volume-extension approach for cases 2-3.

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

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

We conclude that our approach can block misclassifications for cases 2-3. This means that *VSs can be used for reliable classification.*

5 Support Vector Machines

Support Vector Machines (SVMs) construct hyperplanes $h(p, C, \langle I^+, I^- \rangle)$ 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 that controls the trade-off between the margin and the training errors [4].

[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

6 Version Space Support Vector Machines

The key idea of VSSVMs is to use a SVM as a consistency algorithm. By theorem 1 we need a consistency test only for $\langle I^+ \cup \{\mathbf{x}\}, I^-\rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\}\rangle$. Since a SVM is not a consistency algorithm in $H(p)$ [4], we define a hypothesis space $H(p, C, \langle I^+, I^-\rangle)$ for which SVM is a consistency algorithm for $\langle I^+ \cup \{\mathbf{x}\}, I^-\rangle$ and $\langle I^+, I^- \cup \{\mathbf{x}\}\rangle$.

Definition 1 *Given parameters* p *and* C *and data* $\langle I^+, I^-\rangle$*, if* $cons(h(p, C, \langle I^+, I^-\rangle), \langle I^+, I^-\rangle)$*, then* $H(p, C, \langle I^+, I^-\rangle)$ *equals:*

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

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

SVM is a consistency algorithm if the property below holds.

Definition 2 *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* $\langle I^+, I^- \cup \{\mathbf{x}\}\rangle$*.*

Theorem 3 sets the SVM consistency test in $H(p, C, \langle I^+, I^-\rangle)$.

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:*

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

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

Given $H(p, C, \langle I^+, I^-\rangle)$, VSSVMs are defined as follows:

Definition 3 *Given data* $\langle I^{+\prime}, I^{-\prime}\rangle$ *s.t.* $I^{+\prime} \supseteq I^+$ *and* $I^{-\prime} \supseteq I^-$*, the version space support vector machine* $VS_C^p(I^{+\prime}, I^{-\prime})$ *is:*

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

The classification algorithm of VSSVMs realizes the unanimous-voting rule (figure 1). It starts classifying instance $\mathbf{x}$ by first building the SVM hyperplane $h(p, C, \langle I^+, I^-\rangle)$. If $h(p, C, \langle I^+, I^-\rangle)$ is inconsistent with $\langle I^+, I^-\rangle$, by definition 1 $H(p, C, \langle I^+, I^-\rangle) = \emptyset$. Thus, $VS_C^p(I^+, I^-) = \emptyset$ and by the unanimous-voting rule 0 is returned. If $h(p, C, \langle I^+, I^-\rangle)$ is consistent with $\langle I^+, I^-\rangle$, $VS_C^p(I^+, I^-) \neq \emptyset$. Hence, the algorithm tests if $h(p, C, \langle I^+, I^-\rangle)$ is consistent with $\langle I^+ \cup \{\mathbf{x}\}, I^-\rangle$. If so, $VS_C^p(I^+ \cup \{\mathbf{x}\}, I^-) \neq \emptyset$ and the algorithm builds $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\}\rangle)$. If $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\}\rangle)$ is inconsistent with $\langle I^+, I^- \cup \{\mathbf{x}\}\rangle$, 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 class $+1$ is assigned to $\mathbf{x}$. If class $+1$ is not assigned, we check analogously if class -1 can be assigned. If no class is assigned, 0 is returned.

Build a hyperplane $h(p, C, \langle I^+, I^-\rangle)$;
if $\neg cons(h(p, C, \langle I^+, I^-\rangle), \langle I^+, I^-\rangle)$ **then return** 0;
if $cons(h(p, C, \langle I^+, I^-\rangle), \langle I^+ \cup \{\mathbf{x}\}, I^-\rangle)$ **then**
  Build hyperplane $h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\}\rangle)$;
  if $\neg cons(h(p, C, \langle I^+, I^- \cup \{\mathbf{x}\}\rangle), \langle I^+, I^- \cup \{\mathbf{x}\}\rangle)$
  then return $+1$;
if $cons(h(p, C, \langle I^+, I^-\rangle), \langle I^+, I^- \cup \{\mathbf{x}\}\rangle)$ **then**
  Build hyperplane $h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^-\rangle)$;
  if $\neg cons(h(p, C, \langle I^+ \cup \{\mathbf{x}\}, I^-\rangle), \langle I^+ \cup \{\mathbf{x}\}, I^-\rangle)$
  then return -1;
return 0.

Figure 1. The Classification Algorithm of VSSVMs.

The volume-extension approach for VSSVMs is applied for cases 2-3 using the parameter C of SVMs. Note that the probability that the SVM hyperplane $h(p, C, \langle I^+, I^-\rangle)$ is consistent with $\langle I^+, I^-\rangle$ increases with C. Thus, for values C_1 and C_2 of C s.t. $C_1 < C_2$ 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 volume of VSSVMs increases with C.

Applying the approach means to find C s.t. $VS_C^p(I^+, I^-) \subseteq H(p, C, \langle I^+, I^-\rangle)$ classifies instances reliably. Since the volume of VSSVMs increases with C, we can find the minimal value for C in a validation process using binary search s.t. instances are classified reliably (correctly) and the volume of $VS_C^p(I^+, I^-)$ is minimized.

7 Experiments and Comparison

We tested VSSVMs using a polynomial kernel (VSSVM-P). We applied the volume-extension approach by incrementing the parameter C to find the maximal accuracy rate A_m and the maximal coverage rate C_m for A_m (coverage rate is the proportion of classified instances). Table 1 compares VSSVM-P with the naive Bayes classifier (NB) (Bayesian framework) and typicalness-based naive Bayes classifier (TNB) (typicalness framework). The comparison is made using A_m and C_m measured by the leave-one-out method. It shows that VSSVM-P outperform NB and TNB w.r.t. reliable classification.

Data Set	VSSVM-P		NB		TNB	
	C_m	A_m	C_m	A_m	C_m	A_m
labor	0.456	1.0	0.396	1.0	0.351	1.0
hepatitis	0.684	1.0	0.013	1.0	0.013	1.0
breast	0.158	1.0	0.072	1.0	0.079	0.921
sonar	0.221	1.0	0.053	1.0	0.005	1.0
colic	0.079	1.0	0.006	1.0	0.360	0.906
wisc.breast	0.687	1.0	0.017	1.0	0.005	1.0
ionosphere	0.706	1.0	0.168	1.0	0.120	1.0

Table 1. The maximal accuracy A_m and coverage C_m of VSSVM-P, NB, and TNB for 7 datasets from [1].

REFERENCES

[1] C. Blake and C. Merz. UCI repository of ML databases, 1998.
[2] H. Hirsh, N. Mishra, and L. Pitt, 'Version spaces and the consistency problem', *Artificial Intelligence*, **156**(2), 115–138, (2004).
[3] T.M. Mitchell, *Machine learning*, McGraw-Hill, New York, NY, 1997.
[4] V. Vapnik, *Statistical Learning Theory*, John Wiley, NY, 1998.
[5] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*, Springer, New York, NY, 2005.