CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Rescale-invariant SVM for binary classification
Authors	Montazery, Mojtaba;Wilson, Nic
Publication date	2017-08
Original Citation	Montazery, M. and Wilson, N. (2017) 'Rescale-Invariant SVM for Binary Classification', IJCAI'17: Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia 19-25 August, pp. 2501-2507. doi: 10.24963/ijcai.2017/348
Type of publication	Conference item
Link to publisher's version	https://www.ijcai.org/Proceedings/2017 - 10.24963/ijcai.2017/348
Rights	© 2017 International Joint Conferences on Artificial Intelligence
Download date	2026-11-01 19:39:02
Item downloaded from	https://hdl.handle.net/10468/10800



UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Rescale-Invariant SVM for Binary Classification

Mojtaba Montazery and Nic Wilson
Insight Centre for Data Analytics
School of Computer Science and IT
University College Cork, Ireland
{mojtaba.montazery, nic.wilson}@insight-centre.org

Abstract

Support Vector Machines (SVM) are among the best-known machine learning methods, with broad use in different scientific areas. However, one necessary pre-processing phase for SVM is normalization (scaling) of features, since SVM is not invariant to the scales of the features' spaces, i.e., different ways of scaling may lead to different results. We define a more robust decision-making approach for binary classification, in which one sample *strongly* belongs to a class if it belongs to that class for all possible rescalings of features. We derive a way of characterising the approach for binary SVM that allows determining when an instance strongly belongs to a class and when the classification is invariant to rescaling. The characterisation leads to a computational method to determine whether one sample is strongly positive, strongly negative or neither. Our experimental results back up the intuition that being strongly positive suggests stronger confidence that an instance really is positive.

1 Introduction

Among the wide spectrum of (supervised) machine learning techniques, the Support Vector Machine approach [Burges, 1998] is one of the best-known. It is widely used in a variety of application domains from text classification [Tong and Koller, 2001] to biological sciences [Byvatov and Schneider, 2002].

However, scaling of features is a crucial requirement for SVM because, for example, a particular feature with very large values, compared with the other features, might effectively veto the effect of other features on the SVM objective function. Therefore, SVM is clearly sensitive to the way features are scaled [Stolcke *et al.*, 2008; Ben-Hur and Weston, 2010]. The common practice for scalings is based on the properties of input instances [Jain and Dubes, 1988; Aksoy and Haralick, 2001; Tax and Duin, 2000]; as an example of a scaling method, the value of a feature is subtracted by the minimum of all values of that feature in the dataset and divided by the difference between the maximum and minimum. So, the scaling can make the learnt model sometimes highly sensitive to precisely which instances are received. There can also be subjective, and even rather arbitrary, choices in the scaling of the feature spaces; different ways lead to different results.

It is therefore natural to consider a more cautious classification in which an instance is positively (negatively) classified if it is labeled as positive (negative) for all choices of scaling; the other instances, whose classification can depend on the choice of scaling, are labelled as neutral. Thus, this method refines the set of positively (negatively) classified instances by SVM in order to improve the level of confidence in the classification decisions. This could be helpful in certain sensitive decision making applications such as disease diagnosis; e.g., the test for presence of a particular disease would fall into three categories, *positive*, *negative*, and *requires further examination*.

There are some studies attempting to account for the dependence on feature scaling in margin-based optimisation methods from a different perspective, see e.g., [Jebara and Shivaswamy, 2009; Dorkó and Schmid, 2003]. Also, note that the motivation behind this paper is different from studies, like [Xu *et al.*,], which are concerned with improving the robustness of SVM against outliers and noise, since we are specifically focusing on the uncertainty caused by rescaling of features.

The rest of the paper is organised as follows. In Section 2, we rephrase conventional binary SVM to facilitate our extension. Section 3 considers the effect of rescaling and defines strong classification. In Section 4, we characterise rescale-optimality, where the rescale-optimal vectors are those that can be made optimal in the SVM objective function for some rescaling. This characterisation leads to a method for computing strong classification, as described in Section 5. In Section 6, the presented approach is evaluated with 18 benchmarks, which are derived from six real data sets.[1]

[1]Because of the space restrictions, it was not possible to include all the proofs. See `http://ucc.insight-centre.org/nwilson/RescaleInvarSVMClassLonger.pdf` for the missing proofs. The longer document also contains a glossary of symbols.

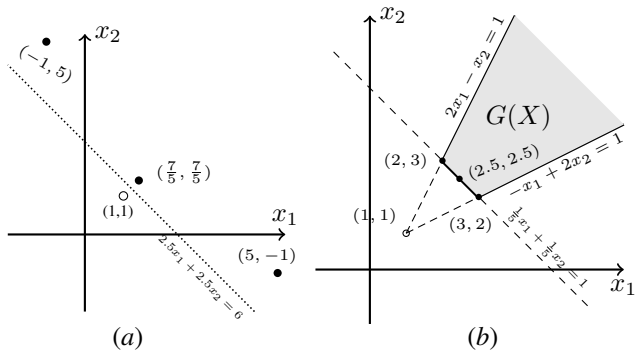


Figure 1: (a) The input samples discussed in Example 1 are shown as black (positive class) and white (negative class) circles. (b) The shaded region shows $G(X)$, with every element of the line segment between $(3, 2)$ and $(2, 3)$ being rescale-optimal in $G(X)$.

2 Standard SVM for Binary Classification

In this section, we introduce some notation, and express standard SVM using that notation, along with some relevant results. This enables easy generalisation to the rescale-invariant case. In this paper, as an initial step, we just consider the case when the training set is consistent (i.e., the instances are linearly separable).

We define the input set X for a binary classification task to be set of samples where each sample is characterised by n features. A sample is expressed as a pair of (x, y), where $x \in I\!R^n$ is the feature vector (i.e., $x(k)$ being the score for x regarding the kth feature[2]), and $y \in \{+1, -1\}$ indicates the class label for that sample. So, $X \subseteq I\!R^n \times \{+1, -1\}$. The input set can be also represented by two disjoint sets:

- $X^+ = \{x^+ : (x^+, +1) \in X\}$
- $X^- = \{x^- : (x^-, -1) \in X\}$

We say that X is *non-trivial* if both X^+ and X^- are non-empty. We define the following terms for any $X \subseteq I\!R^n \times \{+1, -1\}$ which are used throughout the paper, and illustrated in Example 1 below.

- $\Lambda(X) = \{\frac{x^+ - x^-}{2} : x^+ \in X^+,\ x^- \in X^-\}$
- $I(X) = \{1, \ldots, |X^+||X^-|\}$
- $G(X) = \{w \in I\!R^n : \forall \lambda \in \Lambda(X),\ w \cdot \lambda \geq 1\}$
- $MP_w = \min_{x^+ \in X^+} w \cdot x^+$
- $MN_w = \min_{x^- \in X^-} w \cdot (-x^-)$

The dot product $u \cdot v$ is equal to $\sum_{j=1}^{n} u(j)v(j)$ for vectors $u, v \in I\!R^n$.

Example 1. Suppose that $n = 2$, $X^+ = \{(-1, 5), (5, -1), (7/5, 7/5)\}$ and $X^- = \{(1, 1)\}$, with the points marked in Figure 1(a). Then, $\Lambda(X)$ is $\{(-1, 2), (2, -1), (1/5, 1/5)\}$, $I(X) = \{1, 2, 3\}$, and $G(X)$ is the shaded region in Figure 1(b). For $w = (2.5, 2.5)$ (which is in $G(X)$), MP_w and MN_w are 7 and -5 respectively.

By assuming a linear relationship between the feature vector and the class label, each input point $(x, y) \in X$ expresses a linear constraint $y(w \cdot x + b) \geq 1$ with an unknown weight vector $w \in I\!R^n$ and intercept term $b \in I\!R$. Thus, the feasible set $C(X)$ is defined as:

$\{(w, b) \in (I\!R^n, I\!R) : \forall (x, y) \in X, \quad y(w \cdot x + b) \geq 1\}$

The linearity assumption of the model is less restrictive than it sounds; for instance, we could form additional features representing e.g., pairwise products of the basic features, transforming x from $I\!R^n$ to a bigger space (say $\mathcal{H}$). However, in this paper, we assume this transformation has been explicitly defined; we do not consider making use of (non-linear) kernels [Aiserman *et al.*, 1964]. For certain problems the linear kernel works sufficiently well, for instance, when the number of features is large [Hsu *et al.*, 2003] (Appendix C).

The principal idea in SVM [Cortes and Vapnik, 1995] is that from the feasible set $C(X)$ a pair (w, b) is chosen that maximises the margin; this corresponds to the situation when w has the minimum (Euclidean) norm in $\pi(C(X))$, where π is the projection function $\pi : (I\!R^n, I\!R) \to I\!R^n$ given by $\pi(w, b) = w$. We use $\|w\|$ as the notation for Euclidean norm.

We will see in Proposition 1 below that $G(X) = \pi(C(X))$. This fact allows us to make use of general mathematical results from [Wilson and Montazery, 2016a], which considers rescaling in preference learning.

Proposition 1. *Consider any finite and non-trivial $X \subseteq I\!R^n \times \{+1, -1\}$ and $w \in I\!R^n$. Then, $w \in \pi(C(X))$ if and only if $w \in G(X)$. Thus, $\pi(C(X)) = G(X)$.*

As is well-known, the solution that is picked by SVM from $C(X)$ is unique (see e.g., [Burges and Crisp, 1999]). The following theorem restates this fact, making use of our notation.

Theorem 2. *Consider any non-trivial finite $X \subseteq I\!R^n \times \{+1, -1\}$. If $C(X)$ is non-empty then there exists a unique element $(w, b) \in C(X)$ such that w has minimal norm in $\pi(C(X))$. For that unique element, $b = 1 - MP_w = -1 + MN_w$.*

In Example 1, $(2.5, 2.5)$ clearly is the unique element with minimal norm in $G(X)$, and the corresponding b is -6 ($= 1 - MP = -1 + MN$). Thus, its associated hyperplane $2.5x_1 + 2.5x_2 - 6 = 0$, the dotted line in Figure 1(a), produces the maximum margin.

Let us denote the solution of SVM, which by Theorem 2 is unique, by (w^*, b^*), where $b^* = 1 - MP_{w^*} = -1 + MN_{w^*}$. Thereafter, the feature vector $\alpha \in I\!R^n$ with unknown class label is classified as the positive $(+1)$ class label if $w^* \cdot \alpha + b^* \geq 0$, and as the negative (-1) class label otherwise. From now on, we just work with the positive class; the results can be easily applied for the negative class as well.

Theorem 2 leads easily to the following characterisation of positive classification, which is of a form that makes our extension in the following sections more straight-forward.

Proposition 3. *Consider any non-trivial finite $X \subseteq I\!R^n \times \{+1, -1\}$ and any $\alpha \in I\!R^n$. Then, vector α is positively*

[2]Features are assumed to be numeric. However, for ordinal features each value can be replaced by a number, maintaining the order of values. For categorical features one might use the one-hot encoding (a.k.a. 1-of-k coding scheme) to convert a feature with k categories to k Boolean features.

classified if and only if there exists $x^+ \in X^+$ *such that* $w^* \cdot (x^+ - \alpha) \leq 1$.

Proof: Vector α is positively classified if and only if $w^* \cdot \alpha + b^* \geq 0$, which is, by Theorem 2, iff $w^* \cdot \alpha + 1 - MP_{w^*} \geq 0$, i.e., $MP_{w^*} \leq w^* \cdot \alpha + 1$. This holds, from the definition of MP_{w^*}, if and only if $\min_{x^+ \in X^+} w^* \cdot x^+ \leq w^* \cdot \alpha + 1$, which is if and only if there exists $x^+ \in X^+$ such that $w^* \cdot x^+ \leq w^* \cdot \alpha + 1$. This immediately leads to the result. □

3 Rescaling SVM

In this section we consider how performing certain affine transformations, i.e., translations and rescalings, on the domain of each feature, affect the result of SVM. We define a vector as being strongly positively classified, if it is positively classified under all affine transformations of each feature domain.

It is a known fact that the maximum margin hyperplane is essentially unaffected by a translation of feature space (see e.g., [Meila, 2003]); i.e., by moving the origin, the normal vector to the separating hyperplane and hence the result of SVM does not change. Lemma 4 states this formally.

Lemma 4. *Consider any finite and non-trivial* $X \subseteq \mathbb{R}^n \times \{+1, -1\}$*, any* $\alpha, \delta \in \mathbb{R}^n$*, and let* $X^\delta = \{(x + \delta, y) : (x, y) \in X\}$*. Then,* α *is positively classified with respect to* X *if and only if the vector* $\alpha + \delta$ *is positively classified with respect to* X^δ*.*

In contrast with translations, changing the scales of features may significantly affect the result of SVM. Firstly, we define $u \odot v$ to be the vector in $\mathbb{R}^n$ given by pointwise multiplication of $u, v \in \mathbb{R}^n$, and thus, for all $j = 1, \ldots, n$, $(u \odot v)(j) = u(j)v(j)$. Operation $\odot$ is commutative, associative and distributes over addition of vectors, and satisfies the property $(u \odot v) \cdot w = v \cdot (u \odot w)$, for any $u, v, w \in \mathbb{R}^n$. Also, let $\mathbb{R}^n_+$ be the set of strictly positive elements in $\mathbb{R}^n$, i.e., $v \in \mathbb{R}^n$ such that $v(j) > 0$ for all $j = 1, \ldots, n$.

Now, consider the effect of a rescaling $\tau \in \mathbb{R}^n_+$, so that a feature vector $x \in \mathbb{R}^n$ is transformed into $x \odot \tau$. Therefore, X becomes $X_\tau = \{(x \odot \tau, y) : (x, y) \in X\}$. We write the element of $G(X_\tau)$ with minimal norm as w^*_τ.

Example 2. If we rescale X in Example 1 by $\tau = (5, 1)$ then X^+_τ will be $\{(-5, 5), (25, -1), (7\,{}^7\!/_5)\}$, X^-_τ becomes $\{(5, 1)\}$, and consequently, $\Lambda(X_\tau) = \{(-5, 2), (10, -1), (1, {}^1\!/_5)\}$. It can be shown that w^*_τ equals $({}^3\!/_5, 2)$ and $b^*_\tau = -6$. Now, let $\alpha = (-1, 4)$ and so $\alpha \odot \tau = (-5, 4)$. Clearly, when there is no scaling, α is positively classified since $(2.5, 2.5) \cdot (-1, 4) - 6 = 1.5 > 0$, but under scaling τ, $\alpha \odot \tau$ is negatively classified because $({}^3\!/_5, 2) \cdot (-5, 4) - 6 = -1 < 0$.

For a rescaling vector $\tau \in \mathbb{R}^n_+$, we say that $\alpha \in \mathbb{R}^n$ is *positively classified under rescaling* τ iff $\alpha \odot \tau$ is positively classified with respect to X_τ, which, by using Proposition 3, is if and only if there exists $x^+_\tau \in X^+_\tau$ such that $w^*_\tau \cdot (x^+_\tau - (\alpha \odot \tau)) \leq 1$. This holds iff there exists $x^+ \in X^+$ such that $w^*_\tau \cdot ((x^+ \odot \tau) - (\alpha \odot \tau)) \leq 1$, i.e., $(w^*_\tau \odot \tau) \cdot (x^+ - \alpha) \leq 1$.

Let us say that $\alpha \in \mathbb{R}^n$ is *strongly* positively classified if and only if it is positively classified under any rescaling $\tau \in \mathbb{R}^n_+$ and any shift $\delta \in \mathbb{R}^n$. This is if and only if it is positively classified under any rescaling and no shift (i.e., a shift of $\delta = \mathbf{0}$, the zero vector).

Definition 1. For $\alpha \in \mathbb{R}^n$, we define $Y_X(\alpha) = 1$ if and only if α is strongly positively classified; i.e., for all $\tau \in \mathbb{R}^n_+$ there exists $x^+ \in X^+$ such that $(w^*_\tau \odot \tau) \cdot (x^+ - \alpha) \leq 1$.

4 Characterisations Using Rescale Optimality

Here we borrow the notion of rescale-optimality from [Wilson and Montazery, 2016a], and make use of general mathematical results from there. This leads to the computational technique in Section 5 for testing if a vector is strongly positively classified.

We define $\mathrm{RO}(X)$ to be $\{w^*_\tau \odot \tau : \tau \in \mathbb{R}^n_+\}$. Hence, it is clear, from Definition 1, that $Y_X(\alpha) = 1$ if and only if $\forall w \in \mathrm{RO}(X),\ \exists x^+ \in X^+,$ s.t. $w \cdot (x^+ - \alpha) \leq 1$.

Definition 2 (rescale-optimal)**.** For any non-trivial finite $X \subseteq \mathbb{R}^n \times \{+1, -1\}$, and $w \in G(X)$, let us say that w is rescale-optimal in $G(X)$ if there exists $\tau \in \mathbb{R}^n_+$ such that for all $u \in G(X)$, $\|u \odot \tau\|^2 \geq \|w \odot \tau\|^2$.

It can be seen intuitively that every element of the line segment between $(3, 2)$ and $(2, 3)$ in Figure 1(b) is rescale-optimal; if $\tau(1) > \tau(2)$ (i.e., with the ratio $\frac{\tau(1)}{\tau(2)}$ being increased) then $w^*_\tau \odot \tau$ moves from $(2.5, 2.5)$ towards $(3, 2)$. Similarly, increasing the ratio $\frac{\tau(2)}{\tau(1)}$ from 1 moves $w^*_\tau \odot \tau$ from $(2.5, 2.5)$ towards $(2, 3)$. For instance, in Example 2, $\tau(1) = 5\tau(2)$ leads to $w^*_\tau \odot \tau = (3, 2)$.

The following proposition, which follows immediately from Proposition 1 in [Wilson and Montazery, 2016a], states that the set of all rescale-optimal elements of $G(X)$ is $\mathrm{RO}(X)$. As a result of this equivalence, we can say $Y_X(\alpha) = 1$ if and only if for every rescale-optimal element w in $G(X)$, there exists $x^+ \in X^+$ such that $w \cdot (x^+ - \alpha) \leq 1$.

Proposition 5. *Consider any non-trivial finite* $X \subseteq \mathbb{R}^n \times \{+1, -1\}$*, and any* $w \in \mathbb{R}^n$*. Then,* w *is rescale-optimal in* $G(X)$ *if and only if* w *is in* $\mathrm{RO}(X)$*.*

In Section 4.2 below we characterise the rescale-optimal elements, which leads to a computational procedure for strong classification. First, in Section 4.1, we characterise the situations in which rescaling the values of the features makes no difference to the result of the classification, in which case strong classification is the same as the standard classification.

4.1 Determining Invariance to Rescaling

Theorem 6 below shows that rescaling the features vector makes no difference in the classification when there is a unique rescale-optimal vector in $G(X)$ (and the vector thus has minimal norm in $G(X)$).

Theorem 6. *Consider any non-trivial finite* $X \subseteq \mathbb{R}^n \times \{+1, -1\}$*. Let* $Pos(X)$ *be the set of all* $\alpha \in \mathbb{R}^n$ *that are positively classified, and let* $SPos(X)$ *be the set of* α *that are strongly positively classified. Then* $Pos(X) = SPos(X)$ *if and only if there exists a unique rescale-optimal element in* $G(X)$*, i.e.,* $\mathrm{RO}(X)$ *is a singleton set.*

Proof: For $w \in I\!R^n$, let Pos_w be the set of all $\alpha \in I\!R^n$ such that there exists $x^+ \in X^+$ such that $w \cdot (x^+ - \alpha) \leq 1$. Then, by Proposition 3, $Pos(X) = Pos_{w^*}$, and using Proposition 5, $SPos(X)$ is the intersection of Pos_w over all rescale-optimal w in $G(X)$. Note that w^* is rescale optimal in $G(X)$ (using the identity rescaling). Thus, if there is a single rescale-optimal w then $w = w^*$ and so $Pos(X) = SPos(X)$.

To prove the converse, suppose that $Pos(X) = SPos(X)$, which implies that for all rescale-optimal w in $G(X)$, Pos_w contains Pos_{w^*}. Now, we can write Pos_w as $\{\alpha \in I\!R^n : \alpha \cdot w \geq MP_w - 1\}$, and thus Pos_w is a half-space with normal vector w. Hence, since Pos_w contains Pos_{w^*}, there exists real scalar $q > 0$ with $w^* = qw$. Then for any $\tau \in I\!R^n_+$, $\|w^* \odot \tau\|^2 = q^2 \|w \odot \tau\|^2$. By Definition 2, $q = 1$, since both w and w^* are rescale-optimal, and hence $w = w^*$. This shows that there is a unique rescale-optimal element in $G(X)$. □

Definition 3 (pointwise dominance)**.** For $u, w \in I\!R^n$, w pointwise dominates u if $u \neq w$ and for all $j \in \{1, \ldots, n\}$, either $0 \leq w(j) \leq u(j)$ or $0 \geq w(j) \geq u(j)$ (i.e., $w(j)$ is closer to zero than $u(j)$).

Example 3. Consider modifying Example 1 by just removing $(7/5, 7/5)$ from X^+. Then, $\Lambda(X) = \{(-1, 2), (2, -1)\}$, and $G(X)$ becomes the intersection of the half-spaces $-x_1 + 2x_2 \geq 1$ and $2x_1 - x_2 \geq 1$, with a single extremal point $(1, 1)$. This point pointwise dominates every other element in $G(X)$ (see Figure 1(b)). Pointwise dominating every other element, from Theorem 7 below, means that $(1, 1)$ will be the unique element of $G(X)$ that is rescale-optimal. Using Theorem 6, this implies that the classification is invariant to the rescaling (in this example).

Theorem 7 below characterises when there is a unique rescale-optimal element, and Corollary 8 leads to a polynomial algorithm to determine that unique element. These results follow from Theorem 2 and Corollary 1 in [Wilson and Montazery, 2016a; 2016b].

Theorem 7. *Consider any non-trivial finite $X \subseteq I\!R^n \times \{+1, -1\}$ and any $w \in G(X)$. Then, w is the unique element of $G(X)$ that is rescale-optimal if and only if w pointwise dominates every element in $G(X) \setminus \{w\}$.*

Corollary 8. *Consider any non-trivial finite $X \subseteq I\!R^n \times \{+1, -1\}$ and any $v \in G(X)$. Define $w \in I\!R^n$ as follows. For each $j \in \{1, \ldots, n\}$: (i) If $v(j) = 0$ then define $w(j) = 0$. (ii) If $v(j) > 0$ then define $w(j) = \inf\{u(j) : u \in G(X), u(j) \geq 0\}$. (iii) If $v(j) < 0$ then define $w(j) = \sup\{u(j) : u \in G(X), u(j) \leq 0\}$. If $w \in G(X)$ then w is uniquely rescale-optimal in $G(X)$. Also, there exists a uniquely rescale-optimal element in $G(X)$ if and only if $w \in G(X)$.*

4.2 Characterising Rescale-Optimality

Here, we characterise the set of rescale-optimal elements in $G(X)$, which leads to a computational method for strong classification.

Definition 4 (agreeing on signs)**.** For $w, \mu \in I\!R^n$, we say that w *and* μ *agree on signs* if for all $j = 1, \ldots, n$, (i) $w(j) = 0 \iff \mu(j) = 0$; (ii) $w(j) > 0 \iff \mu(j) > 0$; and thus also: (iii) $w(j) < 0 \iff \mu(j) < 0$.

Theorem 9. *Consider any finite non-trivial $X \subseteq I\!R^n \times \{+1, -1\}$, and any non-zero vector $w \in G(X)$. Then, w is rescale-optimal in $G(X)$ if and only if there exists $\mu \in I\!R^n$ and non-negative reals r_i, for each $i \in I(X)$, such that (a) μ agrees on signs with w; (b) $\mu = \sum_{i \in I} r_i \lambda_i$; and (c), for each $i \in I(X)$ either $r_i = 0$ or $\lambda_i \cdot w = 1$, where λ_i is the i^{th} element of $\Lambda(X)$.*

Proof: Theorem 5 of [Wilson and Montazery, 2016a] implies that w is rescale-optimal in $G(X)$ if and only if there exists $\mu \in I\!R^n$ and non-negative reals r_i (for each $i \in I(X)$) such that conditions (a), (b), (c) and (d) hold, where (a), (b) and (c) are the conditions above, and (d) is the condition that $w \cdot \mu = 1$.

$\Rightarrow$: this follows immediately from Theorem 5 of [Wilson and Montazery, 2016a].

$\Leftarrow$: Assume that there exists $\mu \in I\!R^n$ and non-negative reals r_i (for each $i \in I(X)$) such that conditions (a), (b), (c) hold. w is not the zero vector, since $w \in G(X)$. Condition (a), that μ agrees on signs with w, implies that $\mu \cdot w > 0$. Let $\mu' = \frac{\mu}{\mu \cdot w}$, and for each $i \in I(X)$ define $r'_i = \frac{r_i}{\mu \cdot w}$. Then, (a) μ' agrees on signs with w, (b) $\mu' = \sum_{i \in I} r'_i \lambda_i$, (c) for each $i \in I(X)$ either $r'_i = 0$ or $\lambda_i \cdot w = 1$, and (d) $w \cdot \mu' = 1$. Theorem 5 of [Wilson and Montazery, 2016a] then implies that w is rescale-optimal in $G(X)$. □

5 Computation of Strong Classification

Here we express if an instance is strongly positively classified in terms of a set of constraints.

For a set $X \subseteq I\!R^n \times \{+1, -1\}$ and arbitrary $\alpha \in I\!R^n$, we would like to determine if $Y_X(\alpha) = 1$, i.e., if α is strongly positively classified. We can infer from Propositions 3 and 5 that $Y_X(\alpha) \neq 1$ if and only if there exists a rescale-optimal element w in $G(X)$ such that for all $x^+ \in X^+$, $w \cdot (x^+ - \alpha) > 1$. By taking into account Theorem 9 for characterising rescale optimality, we obtain a set of inequalities to determine if $Y_X(\alpha) \neq 1$ as follows.

Let λ_i be the i^{th} element of $\Lambda(X)$ where $i \in I(X)$. Now, $Y_X(\alpha) \neq 1$ if and only if there exists $w \in I\!R^n$ and $\mu \in I\!R^n$, and non-negative reals r_i for each $i \in I(X)$ such that

- $\forall x^+ \in X^+, w \cdot (x^+ - \alpha) > 1$;
- $\forall i \in I(X), w \cdot \lambda_i \geq 1$; (i.e., $w \in G(X)$)
- $\forall i \in I(X), w \cdot \lambda_i = 1$ or $r_i = 0$;
- $\forall j = 1, \ldots, n, w(j) = 0 \iff \mu(j) = 0$, and $w(j) > 0 \iff \mu(j) > 0$; (i.e., agreeing on signs);
- $\mu = \sum_{i \in I(X)} r_i \lambda_i$.

In CPLEX, a disjunctive constraint such as $[w \cdot \lambda_i = 0$ or $r_i = 0]$ can be expressed as $(w \cdot \lambda_i == 0) + (r_i == 0) \geq 1$ (each logical proposition is treated as an integer; 0 for false and 1 for true). The number of constraints here is $2|X^+||X^-| + |X^+| + n + 2$.

Bench.	$\|X\|$	$\|I(X)\|$	Data Set	Features (n)
1.	16	63	Breast Cancer Wisconsin	9
2.	20	84		
3.	25	126		
4.	22	72	Pima Indians Diabetes	8
5.	28	171		
6.	18	65		
7.	20	64	Blood Transfusion	4
8.	25	46		
9.	25	144		
10.	25	84	Indian Liver Patient	10
11.	15	36		
12.	21	108		
13.	30	104	Fertility	9
14.	20	81		
15.	15	14		
16.	25	156	Banknote Authentication	4
17.	20	100		
18.	15	50		

Table 1: The specifications of 18 benchmarks which are used for the experiments; these are derived from six real data sets.

Benchmark	Neutral #	Total #	Ratio(%)
1.	261	683	38
2.	53	100	53
3.	53	100	53
4.	186	200	93
5.	168	200	84
6.	91	100	91
7.	35	100	35
8.	6	100	6
9.	9	100	9
10.	72	100	72
11.	93	100	93
12.	191	200	95
13.	46	70	66
14.	56	81	69
15.	79	85	93
16.	14	100	14
17.	12	100	12
18.	30	100	30
Avg.			55.9

Table 2: The number and ratio of instances labelled as neutral by the rescaling method are shown for each benchmark.

6 Experimental Results

The experiments make use of the *UCI machine learning repository*[3] from which six real data sets are chosen, namely *Breast Cancer Wisconsin* [Street *et al.*, 1993], *Pima Indians Diabetes*, *Blood Transfusion Service Center* [Yeh *et al.*, 2009], *Indian Liver Patient*, *Fertility* [Méndez *et al.*, 2012], and *Banknote Authentication*. In fact, we include data sets that meet three criteria: (i) the data set has only two classes, (ii) the data set consists of only numeric features, and (iii) number of features is at most 10. The first two criteria ensure that the data set complies with the proposed method in this paper, and the third made the computation especially fast. Thereafter, 18 benchmarks are derived from those data sets. Table 1 contains specifications of these benchmarks.

For constructing a benchmark, a random selector creates two disjoint sets from a data set, one for learning (i.e., X) and one for testing. However, a pre-processing phase deletes some elements of the learning set in order to make it consistent (i.e., $C(X) \neq \emptyset$), since in this paper we consider only the consistent case.

For each instance in the testing set (let's say α), the rescaling method determines—based on the learning set X— either (i) it is strongly positive ($Y_X(\alpha) = 1$), or (ii) it is strongly negative ($Y_X(\alpha) = -1$), or (iii) it is neutral ($Y_X(\alpha) \neq 1$ and $Y_X(\alpha) \neq -1$).

The number and percentage of neutral instances for each benchmark can be found in Table 2. The ratio of neutral instances for the benchmarks (i.e., those instances being classified differently under different rescaling) varies from 6% to 95% with a mean of 55.9%. For more than half of the instances, rescaling the feature values can change the result of the SVM classification. That points out the unreliability of standard SVM–specifically for larger numbers like 95%-– with respect to being sensitive to the way the features are scaled.

In a testing set, among the non-neutral instances, we can also count:

1. The number of negative instances improperly *strongly* classified as positive (False Positive).
2. The number of positive instances properly *strongly* classified as positive (True Positive).
3. The number of positive instances improperly *strongly* classified as negative (False Negative).
4. The number of negative instances properly *strongly* classified as negative (True Negative).

Conventional SVM can also predict a class label for each instance, positive or negative. As a result, we have False Positive (FP), True Positive (TP), False Negative (FN) and True Negative (TN) for SVM as well. Tables 3 and 4 compare these measures between the two methods. Note that the value of FP + TN (resp. FN + TP) for the rescaling method is not (necessarily) equal to the total number of testing instances with real negative (resp. positive) class label because some instances are classified as neutral.

In Tables 3 and 4, we also compare the *Positive Predictive Value* (PPV) and *Negative Predictive Value* (NPV) of the two methods. PPV is the fraction of truly positive instances among positively classified ones. Similarly, NPV is the fraction of truly negative instances among negatively classified ones. Thus, PPV equals $\frac{\text{TP}}{\text{FP+TP}}$ and NPV equals $\frac{\text{TN}}{\text{FN+TN}}$. The measures PPV and NPV are widely used in medical contexts in order to validate a disease diagnostic test [Parikh *et al.*, 2008]. To illustrate, PPV expresses that if a patient's test is positive how likely it is that he really has the disease. Similarly, NPV means that if a patient's test is negative how likely

[3] http://archive.ics.uci.edu/ml/

Bench.	SVM			Rescaling Method		
	FP	TP	PPV(%)	FP	TP	PPV(%)
1.	15	439	97	0	333	**100**
2.	3	71	96	0	46	**100**
3.	9	64	88	0	47	**100**
4.	25	48	**66**	1	1	50
5.	38	53	58	1	2	**67**
6.	8	15	65	0	0	-
7.	25	6	19	5	4	**44**
8.	7	4	**43**	4	3	36
9.	27	9	**25**	25	8	24
10.	15	13	46	0	2	**100**
11.	35	15	**30**	1	0	0
12.	44	23	34	3	4	**57**
13.	15	2	**12**	1	0	0
14.	19	1	5	3	1	**25**
15.	11	3	21	0	0	-
16.	3	45	**94**	3	38	93
17.	1	42	**98**	1	38	97
18.	3	44	94	0	32	**100**
Avg.			56.56			**62.06**

Table 3: A comparison, using 18 benchmarks, between the PPV of SVM and the rescaling method. The undefined values, labelled -, are excluded from the mean.

Bench.	SVM			Rescaling Method		
	FN	TN	NPV(%)	FN	TN	NPV(%)
1.	10	219	96	3	86	**97**
2.	1	25	96	0	1	**100**
3.	0	27	100	0	0	-
4.	22	105	83	0	12	**100**
5.	14	95	87	0	29	**100**
6.	21	56	73	0	9	**100**
7.	17	52	75	11	45	**80**
8.	18	71	80	18	69	**79**
9.	9	55	**86**	8	50	**86**
10.	19	53	**74**	9	17	65
11.	13	37	**74**	2	4	67
12.	27	106	80	3	4	**100**
13.	6	47	89	1	22	**96**
14.	6	55	90	1	20	**95**
15.	8	63	89	0	6	**100**
16.	2	50	96	0	45	**100**
17.	1	56	98	0	49	**100**
18.	2	51	96	0	38	**100**
Avg.			86.00			**91.47**

Table 4: A comparison, using 18 benchmarks, between the NPV of SVM and the rescaling method. The undefined values, labelled -, are excluded from the mean.

it is that she really doesn't have the disease. The results show an increase of 5.5% in PPV and NPV for the benchmarks on average.

Note that our approach is not intended to be a competitor of SVM in terms of classification accuracy. What it does is to highlight certain instances where we can have greater confidence, with the other instances having reduced confidence because the result of the classification could be changed if a different scaling of the features were used. This is why PPV and NPV are appropriate measures for the experimental results.

The approach discussed in Section 5 was implemented using the solver CPLEX 12.6.2. Determining whether an instance is strongly positive or strongly negative or neutral for any of benchmarks takes less than a couple of seconds, making use of a computer facilitated by a Core i7 2.60 GHz processor and 8 GB RAM memory.

7 Discussion

The scaling of individual features, before applying an SVM method, can be subjective and arbitrary. However, the way features are scaled can make a difference; in fact, for a little more than half of the instances in our experiments, rescaling the feature values can change the result of the SVM classification. Based on this fact, we say an instance strongly belongs to a class if it belongs to that class for all rescalings of features. This new definition boosts the confidence of labeling instances by excluding those instances which are classified differently under different scalings.

Building on the general mathematical results of [Wilson and Montazery, 2016a], we have developed a computational procedure that can test if an instance is strongly positive, i.e., labelled positive for every affine way of rescaling the values of each feature (and similarly for the negative case). We also have a polynomial algorithm for determining when rescaling makes no difference, i.e., when, for a given training set, the classification of any possible instance is not affected by rescaling. Our experiments also showed a slight improvement in the value of prediction, i.e., the likelihood that if an instance is classified as a particular class, it truly belongs to that class.

There are many potential future directions following from this work; for instance, extending the method to the case when there is inconsistency in the input data (i.e., soft margin SVM); attempting to develop the computational method for certain kernels; performing multi-class classification by making use of repeated binary-class classification (e.g., using one-vs-all or one-vs-one approaches); and computing variations of the method where the user can limit the rescaling range for different features.

Acknowledgments

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289. Thanks to the reviewers for their comments, which helped improve the final version of the paper.

References

[Aiserman *et al.*, 1964] MA Aiserman, EM Braverman, and LI Rozonoer. Theoretical foundations of the potential function method in pattern recognition. *Avtomat. i Telemeh*, 25:917–936, 1964.

[Aksoy and Haralick, 2001] Selim Aksoy and Robert M Haralick. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern recognition letters*, 22(5):563–582, 2001.

[Ben-Hur and Weston, 2010] Asa Ben-Hur and Jason Weston. A users guide to support vector machines. *Data mining techniques for the life sciences*, pages 223–239, 2010.

[Burges and Crisp, 1999] Christopher JC Burges and David J Crisp. Uniqueness of the SVM solution. In *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, volume 99, pages 223–229, 1999.

[Burges, 1998] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[Byvatov and Schneider, 2002] Evgeny Byvatov and Gisbert Schneider. Support vector machine applications in bioinformatics. *Applied bioinformatics*, 2(2):67–77, 2002.

[Cortes and Vapnik, 1995] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[Dorkó and Schmid, 2003] Gyuri Dorkó and Cordelia Schmid. Selection of scale-invariant parts for object class recognition. In *9th IEEE International Conference on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France*, volume 1, pages 634–640, 2003.

[Hsu *et al.*, 2003] Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. A practical guide to support vector classification, 2003.

[Jain and Dubes, 1988] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[Jebara and Shivaswamy, 2009] Tony Jebara and Pannagadatta K Shivaswamy. Relative margin machines. In *Advances in Neural Information Processing Systems*, pages 1481–1488, 2009.

[Meila, 2003] Marina Meila. Data centering in feature space. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics, AISTATS 2003, Key West, Florida, USA, January 3-6, 2003*, 2003.

[Méndez *et al.*, 2012] David G Méndez, Jose L Girela, Joaquin De Juan, M Jose Gomez-Torres, and Magnus Johnsson. Predicting seminal quality with artificial intelligence methods. *Expert Systems with Applications*, 39(16):12564–12573, 2012.

[Parikh *et al.*, 2008] Rajul Parikh, Annie Mathai, Shefali Parikh, G Chandra Sekhar, Ravi Thomas, et al. Understanding and using sensitivity, specificity and predictive values. *Indian journal of ophthalmology*, 56(1):45, 2008.

[Stolcke *et al.*, 2008] Andreas Stolcke, Sachin Kajarekar, and Luciana Ferrer. Nonparametric feature normalization for SVM-based speaker verification. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2008, March 30 - April 4, 2008, Caesars Palace, Las Vegas, Nevada, USA*, pages 1577–1580. IEEE, 2008.

[Street *et al.*, 1993] W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. In *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, pages 861–870. International Society for Optics and Photonics, 1993.

[Tax and Duin, 2000] DM Tax and Robert PW Duin. Feature scaling in support vector data descriptions. *Learning from Imbalanced Datasets*, pages 25–30, 2000.

[Tong and Koller, 2001] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *Journal of machine learning research*, 2(Nov):45–66, 2001.

[Wilson and Montazery, 2016a] Nic Wilson and Mojtaba Montazery. Preference inference through rescaling preference learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2203–2209. IJCAI/AAAI Press, 2016.

[Wilson and Montazery, 2016b] Nic Wilson and Mojtaba Montazery. *Preference Inference Through Rescaling Preference Learning (extended version of IJCAI 2016 paper including proofs)*. Available at http://ucc.insight-centre.org/nwilson/RescalingProofs.pdf, 2016.

[Xu *et al.*,] Linli Xu, Koby Crammer, and Dale Schuurmans. Robust support vector machine training via convex outlier ablation. In *Proceedings of The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*.

[Yeh *et al.*, 2009] I-Cheng Yeh, King-Jang Yang, and Tao-Ming Ting. Knowledge discovery on RFM model using Bernoulli sequence. *Expert Systems with Applications*, 36(3):5866–5871, 2009.