ELSEVIER

Pattern Recognition Letters 22 (2001) 1263–1272

Pattern Recognition Letters

www.elsevier.com/locate/patrec

Support vector machines with different norms: motivation, formulations and results

João Pedro Pedroso [a,*], Noboru Murata [b,1]

[a] *Centro de Investigação Operacional, Faculdade de Ciências da Universidade de Lisboa, Campo Grande, 1749-016 Lisboa, Portugal*
[b] *Department of Electrical, Electronics, and Computer Engineering, School of Science and Engineering, Waseda University, 3-4-1 Ohkubo, Shinjuku, Tokyo 169-8555, Japan*

Abstract

We introduce two formulations for training support vector machines, based on considering the L_1 and L_∞ norms instead of the currently used L_2 norm, and maximising the margin between the separating hyperplane and each data sets using L_1 and L_∞ distances. We exploit the geometrical properties of these different norms, and propose what kind of results should be expected for them. Formulations in mathematical programming for linear problems corresponding to L_1 and L_∞ norms are also provided, for both the separable and non-separable cases. We report results obtained for some standard benchmark problems, which confirmed that the performance of all the formulations is similar. As expected, the CPU time required for machines solvable with linear programming is much shorter.

Keywords: Support vector machines; Linear programming

1. Introduction

Linear programming approaches to support vector machines have recently been addressed (Bennett and Bredensteiner, 2000; Bradley and Mangasarian, 1998; Smola et al., 1998). Their main advantage concerns the possibility of using solvers for linear problems, with improved reliability and speed as compared to solvers for quadratic problems, and also with the capacity of solving very large size problems.

The initial formulation of the problem of optimal separation of two classes of points can be found, for example, in (Vapnik, 1995; Burges, 1998). It consists in finding the hyperplane that separates the two sets in such a way that the (Euclidean) distance between the hyperplane and nearest point of each of the data sets is maximum. In this paper, we will adopt the same approach, but we will replace the Euclidean (L_2) distance by the L_1 norm and L_∞ norm distances. This will allow us to write a version of the problem which shares the theoretical advantages of original formulation, but that is much simpler to tackle.

1.1. Standard formulation

Let us label the training data $x_i \in \mathscr{R}^d$ with a label $y_i \in \{-1, +1\}$, for all the training examples $i = 1, \ldots, l$, where l is the number of examples, d is

* Corresponding author. Fax: +351-217-500-081.
E-mail addresses: jpp@fc.ul.pt (J.P. Pedroso), murata@elec.waseda.ac.jp (N. Murata).
[1] Fax: +81-3-5286-3383.

0167-8655/01/$ - see front matter
PII: S0167-8655(01)00071-X

the number of discriminating attributes (i.e., the dimension of the problem). Let us call positive (negative) examples those for which $y_i = +1$ ($y_i = -1$). The set of positive examples is denoted by $\mathscr{P}$, and the set of negative examples is denoted by $\mathscr{N}$.

1.1.1. Separable case

Admitting that there exists a hyperplane $H_{\boldsymbol{w},b} : \boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ separating positive from negative examples, the separation problem is to determine the hyperplane such that $\boldsymbol{w} \cdot \boldsymbol{x}_i + b \geqslant +1$ for positive examples, $\boldsymbol{w} \cdot \boldsymbol{x}_i + b \leqslant -1$ for negative examples, with maximal distance to the closest point of each of the classes. The objective is:

$$\underset{\boldsymbol{w},b}{\text{maximise}} \frac{2}{\|\boldsymbol{w}\|}. \tag{1}$$

This problem can equivalently be solved by minimising $\|\boldsymbol{w}\|^2/2$.

1.1.2. Non-separable case

If the problem stated above has no feasible solution, the constraints can be relaxed, and a penalty added to the objective function (using the so-called *soft constraints*). In this case, the constraints become $\boldsymbol{w} \cdot \boldsymbol{x}_i + b \geqslant +1 - \xi_i$ for positive examples, $\boldsymbol{w} \cdot \boldsymbol{x}_i + b \leqslant -1 + \xi_i$ for negative examples, with $\xi_i \geqslant 0$. The objective function becomes

$$\underset{\boldsymbol{w},b,\xi}{\text{minimise}} \left\{ \frac{\|\boldsymbol{w}\|^2}{2} + C\left(\sum_i \xi_i \right)^k \right\}, \tag{2}$$

where C is a user-defined parameter that controls the importance of the classification errors in the training process, and k is a positive integer.

2. Mathematical background

2.1. Conjugate property

Let us think of two parallel hyperplanes in $\mathbb{R}^d$, $H_1 : \boldsymbol{w} \cdot \boldsymbol{x} + b_1 = 0$ and $H_2 : \boldsymbol{w} \cdot \boldsymbol{x} + b_2 = 0$. The distance with L_p norm between these two hyperplanes is defined as

$$d_p(H_1, H_2) \overset{\text{def.}}{=} \min_{\substack{\boldsymbol{x} \in H_1 \\ \boldsymbol{y} \in H_2}} \|\boldsymbol{x} - \boldsymbol{y}\|_p, \tag{3}$$

where

$$\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}. \tag{4}$$

Taking an arbitrary point $\boldsymbol{y} \in H_2$, we can rewrite this distance as: $d_p(H_1, H_2) = \min_{x \in H_1} \|\boldsymbol{x} - \boldsymbol{y}\|_p$. Shifting both hyperplanes parallelly so that H_2 passes through the origin, we obtain two hyperplanes separated the same distance, whose equations are $H_1' : \boldsymbol{w} \cdot \boldsymbol{x} + (b_1 - b_2) = 0$ and $H_2' : \boldsymbol{w} \cdot \boldsymbol{x} = 0$. If we choose point $\boldsymbol{y}$ to be the origin, the distance between these hyperplanes is determined by

$$d_p(H_1', H_2') = \min_{\boldsymbol{x} \in H_1'} \|\boldsymbol{x}\|_p. \tag{5}$$

The so-called *conjugate norm* of L_p is L_q, where p and q satisfy the equality

$$\frac{1}{p} + \frac{1}{q} = 1. \tag{6}$$

The Hölder inequality (see Appendix A) states that for conjugate norms L_p and L_q the following inequality holds:

$$\|\boldsymbol{x}\|_p \|\boldsymbol{w}\|_q \geqslant |\boldsymbol{x} \cdot \boldsymbol{w}|. \tag{7}$$

For $\boldsymbol{x} \in H_1'$, the right term of this inequality is constant, and its value is given by $|\boldsymbol{x} \cdot \boldsymbol{w}| = |b_1 - b_2|$. Therefore, $\min_{x \in H_1'} \|\boldsymbol{x}\|_p \|\boldsymbol{w}\|_q = |b_1 - b_2|$. Hence, we can write the distance between two hyperplanes using the L_p norm as

$$d_p(H_1, H_2) = \min_{\boldsymbol{x} \in H_1'} \|\boldsymbol{x}\|_p = \frac{|b_1 - b_2|}{\|\boldsymbol{w}\|_q}, \tag{8}$$

where q is the conjugate norm of p.

2.2. Duality of observations and parameters

Using distances as determined by the L_p norm, we can define the margin in L_p norm sense. Going back to the classification problem, let $\boldsymbol{y}$ be a training example and $H : \boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ be a separating hyperplane.

A hyperplane H' which is parallel to H and passes through $\boldsymbol{y}$ is defined by the equation $H' : \boldsymbol{w} \cdot \boldsymbol{x} - \boldsymbol{w} \cdot \boldsymbol{y} = 0$. With the L_p norm distance,

the margin between the separating hyperplane H and the training example $\boldsymbol{y}$ is defined as

$$\begin{aligned}\text{margin}_H(\boldsymbol{y}) &\stackrel{\text{def.}}{=} \min_{\boldsymbol{x}\in H} \|\boldsymbol{x}-\boldsymbol{y}\|_p = d_p(H,H') \\ &= \frac{|b+\boldsymbol{w}\cdot\boldsymbol{y}|}{\|\boldsymbol{w}\|_q}. \end{aligned} \tag{9}$$

If we consider the space of the parameters $\boldsymbol{w}$ and b, an example $\boldsymbol{y}$ defines a hyperplane $Y : \boldsymbol{w}\cdot\boldsymbol{y}+b=0$ which divides the whole parameter space into two parts: one where all $\boldsymbol{w}, b$ correctly classify the example $\boldsymbol{y}$, and the other where no parameters $\boldsymbol{w}, b$ correctly classify $\boldsymbol{y}$. The intersection of regions where the parameters correctly classify all the examples is called the *version space*.

For the classification task, only the sign of $\boldsymbol{w}\cdot\boldsymbol{x}+b$ is important; the absolute value of this quantity is irrelevant. Therefore, a constraint for scaling $(\boldsymbol{w}, b)$ is usually added. A simple constraint, which is convenient for the purpose of numerical simplification, is to set $|\boldsymbol{w}\cdot\boldsymbol{y}+b| \geqslant +1$, and solving

$$\text{maximise} \frac{2}{\|\boldsymbol{w}\|_q}, \tag{10}$$

which is a natural extension of the formulation stated in Section 1. This will be used for the derivation of the L_1 norm and L_∞ norm support vector machines in the following sections.

Here we will consider another constraint, which allows us to interpret what happens to the version space when we maximise the margin in the original space (the *feature space*). We will set $\|\boldsymbol{w}\|_q = 1$; considering this constraint, the problem is reduced to maximising the minimum margin between the separating hyperplane and the examples, for all the training examples. This leads to the problem $\text{maximise}_{\boldsymbol{w},b} \min_i \text{margin}_{H(\boldsymbol{w},b)}(\boldsymbol{y}_i)$, which is equivalent to

$$\underset{\boldsymbol{w},b}{\text{maximise}} \min_i |b+\boldsymbol{w}\cdot\boldsymbol{y}_i|. \tag{11}$$

In the case of L_2 norm, the so-constrained parameter space of $\boldsymbol{w}$ is a hypersphere, and in the case of L_1 norm and L_∞ norm it is a hypercubic. It is easy to understand the underlying geometrical structure: we aim at finding the point in the hypersphere or hypercube that is most distant from the closest hyperplane delimiting the version space. See Section 5 for more details and examples on this.

3. L_1 norm formulation

When L_1 norm is used for determining the distances between the training points and the separating hyperplane, the problem can be formulated as a linear program.

Let us first recall that the distance between two parallel hyperplanes $H_1 : \boldsymbol{w}\cdot\boldsymbol{x}+b_1=0$ and $H_2 : \boldsymbol{w}\cdot\boldsymbol{x}+b_2=0$ using the L_1 norm is given by

$$d_1(H_1,H_2) = \frac{|b_1-b_2|}{\|\boldsymbol{w}\|_\infty}. \tag{12}$$

3.1. Separable case

If we impose that $\boldsymbol{w}\cdot\boldsymbol{x}_i+b=\pm 1$ for support vectors (this corresponds to scaling $(\boldsymbol{w}, b)$), then the distance between the two "support hyperplanes" that go through the support vectors of each class is

$$d_1(H^+,H^-) = \frac{|(b+1)-(b-1)|}{\|\boldsymbol{w}\|_\infty} = \frac{2}{\max_j |w_j|}. \tag{13}$$

We want to maximise this value, what is equivalent to the following minimisation:

$$\underset{\boldsymbol{w},b}{\text{minimise}} \max_j |w_j|. \tag{14}$$

This can be done through linear optimisation, if we add an auxiliary variable a, and two sets of constraints: $a \geqslant w_j\ \forall j$ and $a \geqslant -w_j\ \forall j$. Our objective is now to minimise a subject to all the constraints, as stated in the following linear program:

$$\begin{aligned} \text{minimise} \quad & a \\ \text{subject to} \quad & \boldsymbol{w}\cdot\boldsymbol{x}_i+b \geqslant 1 \quad \forall i\in\mathscr{P}, \\ & \boldsymbol{w}\cdot\boldsymbol{x}_i+b \leqslant -1 \quad \forall i\in\mathscr{N}, \\ & a \geqslant w_j \quad \forall j\in\{1,\ldots,d\}, \\ & a \geqslant -w_j \quad \forall j\in\{1,\ldots,d\}, \\ & a,b\in\mathbb{R}, \boldsymbol{w}\in\mathbb{R}^d. \end{aligned} \tag{15}$$

The support vectors are the training data points for which $\boldsymbol{w}\cdot\boldsymbol{x}_i + b = \pm 1$, i.e., the points for which the corresponding inequalities in (15) are binding. These points can be easily determined, as standard linear programming software outputs values of the slack and dual variables for each constraint.

3.2. Non-separable case

When the data are non-separable, we can use soft-margin constraints and include a penalty in the objective function, as in the case of L_2 norm. For the formulation to be possible with a linear program, we must set $k = 1$ in Eq. (2). The constraints become then $\boldsymbol{w}\cdot\boldsymbol{x}_i + b \geqslant +1 - \xi_i$ for positive examples, $\boldsymbol{w}\cdot\boldsymbol{x}_i + b \leqslant -1 + \xi_i$ for negative examples, with $\xi_i \geqslant 0$. The objective is now

$$\underset{\boldsymbol{w},b,\xi}{\text{minimise}} \left\{ \max_j |w_j| + C\sum_i \xi_i \right\}. \tag{16}$$

For solving this problem using linear programming, if we use an auxiliary variable a that gets the value of the largest $|w_j|$, as stated in the previous section, one formulation is:

$$\begin{aligned} \text{minimise} \quad & a + C\left(\sum_{i\in\mathscr{P}} \xi_i^+ + \sum_{i\in\mathscr{N}} \xi_i^-\right) \\ \text{subject to} \quad & \boldsymbol{w}\cdot\boldsymbol{x}_i + b \geqslant 1 - \xi_i^+ \quad \forall i \in \mathscr{P}, \\ & \boldsymbol{w}\cdot\boldsymbol{x}_i + b \leqslant -1 + \xi_i^- \quad \forall i \in \mathscr{N}, \\ & a \geqslant w_j \quad \forall j \in \{1,\ldots,d\}, \\ & a \geqslant -w_j \quad \forall j \in \{1,\ldots,d\}, \\ & a, b \in \mathbb{R}, \boldsymbol{w} \in \mathbb{R}^d, \xi^+ \in \mathbb{R}_{0+}^{|\mathscr{P}|}, \xi^- \in \mathbb{R}_{0+}^{|\mathscr{N}|}. \end{aligned} \tag{17}$$

4. L_∞ norm formulation

If one uses the L_∞ norm for determining the distances between the training points and the separating hyperplane, the problem can also be formulated as a linear optimisation problem.

Let us recall that using the L_∞ norm, the distance between two parallel hyperplanes $H_1 : \boldsymbol{w}\cdot\boldsymbol{x} + b_1 = 0$ and $H_2 : \boldsymbol{w}\cdot\boldsymbol{x} + b_2 = 0$ is given by

$$d_\infty(H_1, H_2) = \frac{|b_1 - b_2|}{\|\boldsymbol{w}\|_1}. \tag{18}$$

4.1. Separable case

As for the case of L_1 norm, we may impose that $\boldsymbol{w}\cdot\boldsymbol{x}_i + b = \pm 1$ for support vectors. The distance between the two support hyperplanes that go through the support vectors of each class is then

$$d_\infty(H^+, H^-) = \frac{|(1-b) - (-1-b)|}{\|\boldsymbol{w}\|_1} = \frac{2}{\sum_j |w_j|}. \tag{19}$$

We want to maximise this value; this is equivalent to

$$\underset{\boldsymbol{w},b}{\text{minimise}} \sum_j |w_j|. \tag{20}$$

This, again, can be done through linear optimisation. If in the formulation we add, for each w_j, an auxiliary variable a_j, and the constraints $a_j \geqslant w_j\ \forall j$, and $a_j \geqslant -w_j\ \forall j$, then the linear problem is to minimise $\sum_j a_j$, subject to all the constraints. The complete linear programming formulation is the following:

$$\begin{aligned} \text{minimise} \quad & \sum_{j=1}^{d} a_j \\ \text{subject to} \quad & \boldsymbol{w}\cdot\boldsymbol{x}_i + b \geqslant 1 \quad \forall i \in \mathscr{P}, \\ & \boldsymbol{w}\cdot\boldsymbol{x}_i + b \leqslant -1 \quad \forall i \in \mathscr{N}, \\ & a_j \geqslant w_j \quad \forall j \in \{1,\ldots,d\}, \\ & a_j \geqslant -w_j \quad \forall j \in \{1,\ldots,d\}, \\ & b \in \mathbb{R}, \boldsymbol{a}, \boldsymbol{w} \in \mathbb{R}^d. \end{aligned} \tag{21}$$

4.2. Non-separable case

This case is similar to the L_2 and L_1 cases; we use soft-margin constraints with the following objective:

$$\underset{\boldsymbol{w},b,\xi}{\text{minimise}}\left\{\sum_j |w_j| + C\sum_i \xi_i\right\}. \tag{22}$$

This can still be solved using linear programming, if we use the following formulation:

$$\begin{aligned}
\text{minimise}\quad & \sum_{j=1}^{d} a_j + C\left(\sum_{i\in\mathscr{P}} \xi_i^+ + \sum_{i\in\mathscr{N}} \xi_i^-\right)\\
\text{subject to}\quad & \boldsymbol{w}\cdot\boldsymbol{x}_i + b \geqslant 1 - \xi_i^+ \quad \forall i \in \mathscr{P},\\
& \boldsymbol{w}\cdot\boldsymbol{x}_i + b \leqslant -1 + \xi_i^- \quad \forall i \in \mathscr{N},\\
& a_j \geqslant w_j \quad \forall j \in \{1,\ldots,d\},\\
& a_j \geqslant -w_j \quad \forall j \in \{1,\ldots,d\},\\
& b \in \mathbb{R}, \boldsymbol{a}, \boldsymbol{w} \in \mathbb{R}^d, \xi^+ \in \mathbb{R}_{0+}^{|\mathscr{P}|}, \xi^- \in \mathbb{R}_{0+}^{|\mathscr{N}|}.
\end{aligned} \tag{23}$$

5. Visualisation

5.1. Feature space

We try here to make a quick comparison between the type of distances that are considered in this paper. Let us consider two arbitrary points, for the sake of simplicity in the two-dimension Euclidean space (see Fig. 1).

We can think of the L_2 norm distance as the distance from one point to the other in the (arbitrary) slope defined by these points. With the L_1 norm, we can measure the distance considering that we could move only horizontally or vertically from one point to another, and summing the horizontal and the vertical movements. With the L_∞ norm, again we can move only horizontally or vertically from one point to another, but only the biggest of the horizontal or vertical distances matters. We can easily see that the "paths" for measuring each of the distances form a right-angled triangle; the L_2 norm distance is the size of the hypotenuse, the L_1 norm distance is the sum of the sizes of the cathetus, and the L_∞ norm distance is the largest cathetus. Clearly, for any points $\boldsymbol{a}, \boldsymbol{b}$ the inequality between the distances in the three different norms, $\|\boldsymbol{a}-\boldsymbol{b}\|_\infty \leqslant \|\boldsymbol{a}-\boldsymbol{b}\|_2 \leqslant \|\boldsymbol{a}-\boldsymbol{b}\|_1$, holds; therefore, we can think of the L_2 norm as something between the L_∞ norm and the L_1 norm. Note that for points in a line parallel to one of the axes the preceding equation comes to $d_\infty(\boldsymbol{a},\boldsymbol{b}) = d_2(\boldsymbol{a},\boldsymbol{b}) = d_1(\boldsymbol{a},\boldsymbol{b})$.

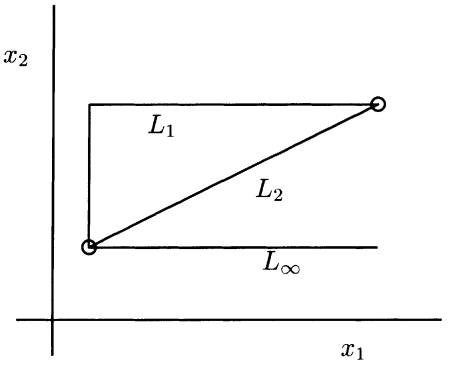

Fig. 1. Distances for L_1, L_2, and L_∞ norms.

Turning back to the support vector machines, we can expect that L_2 norm gives, in the separable case, a result that is "between" L_1 norm and L_∞ norm. Intuitively, we can see that for a given classification problem, the L_∞ norm support vector machine will be closer to a parallel of one of the axes than the L_2 norm machine; and the L_1 norm machine will be closer to a 45 degree hyperplane. This can be visualised in Fig. 2.

Another observation that we can make for the support vector machines considered in this paper is that as the number of training examples gets

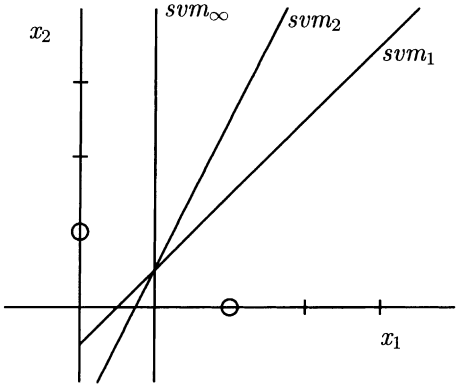

Fig. 2. Support vector machines obtained for L_1, L_2, and L_∞ norms, when there are two data points to classify: (0, 1) for one class and (2, 0) for the other.

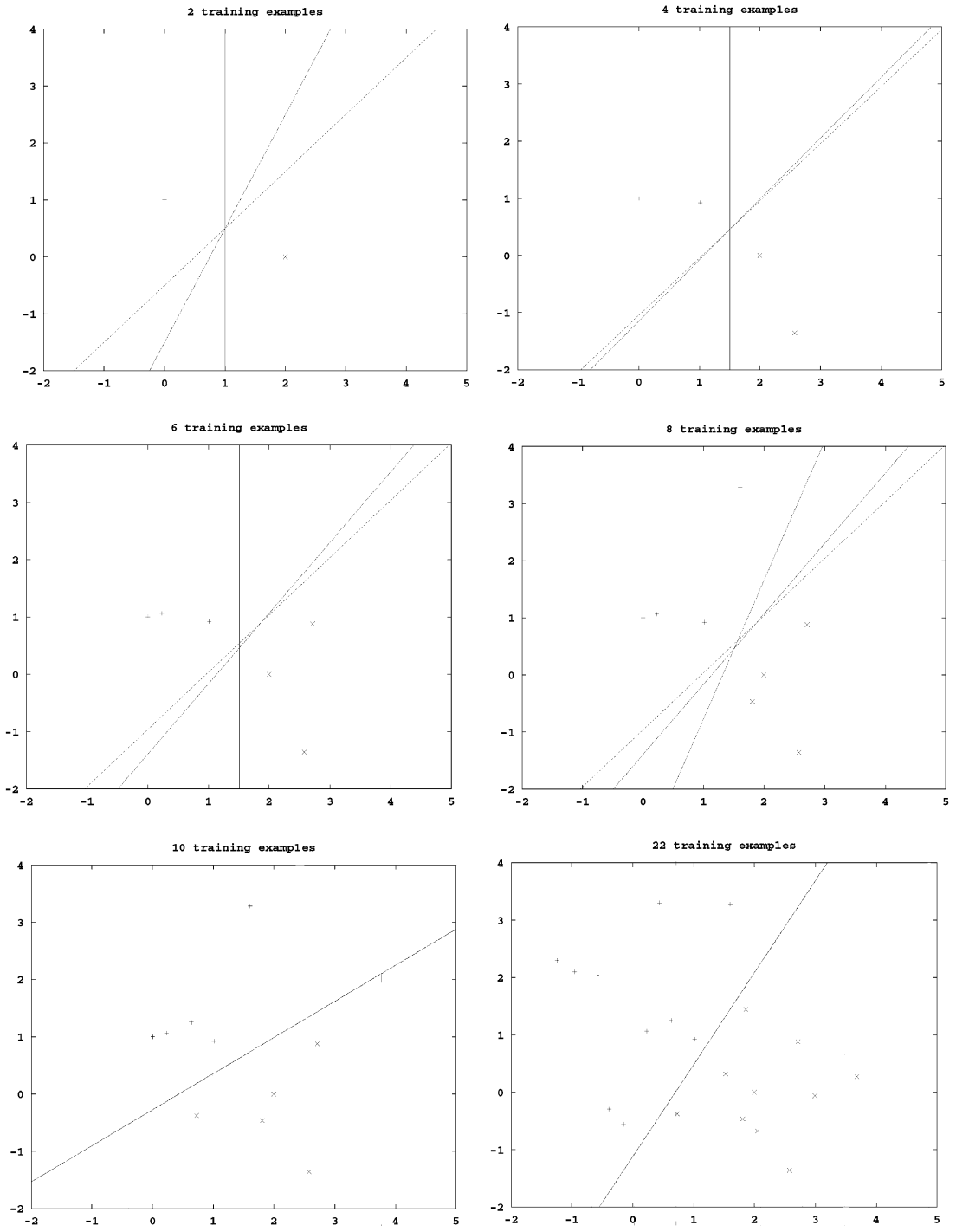

Fig. 3. Plots for increasing sizes of random data points added to (0, 1)/(2, 0). Solid line corresponds to L_∞ norm solution, dashed line to L_2 norm, and dotted line to L_1 norm.

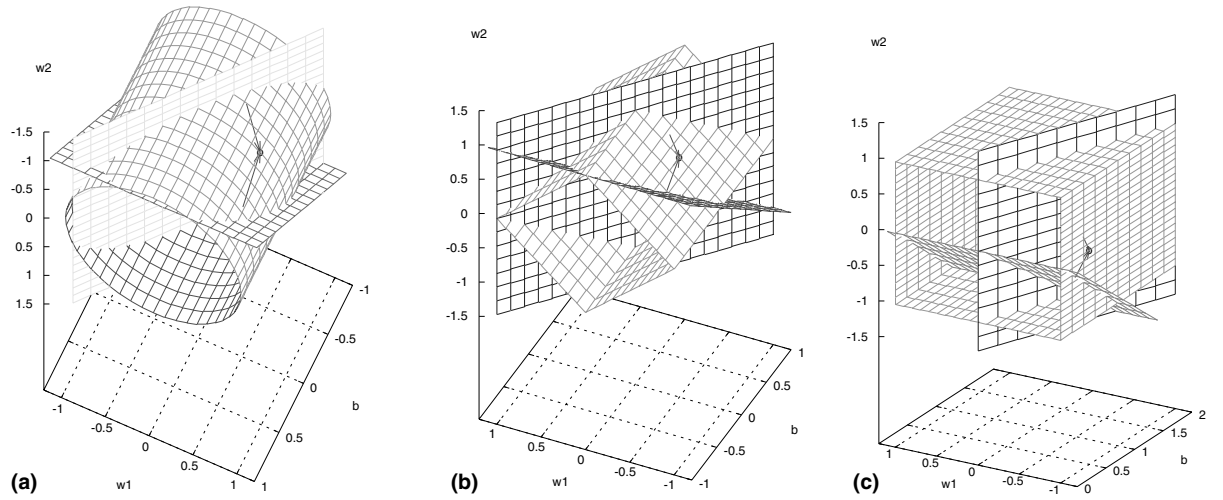

Fig. 4. Version space: visualisation of what happens in the case of L_2 norm (a), L_1 norm (b), and L_∞ norm (c). The hyperplanes correspond to given training examples, and limit the feasible region to the points on one of their sides. The cylinder and prisms correspond to the constraint that $\|\boldsymbol{w}\| = 1$ for each of the norms considered. The problem is to find the point in the surface of the cylinder or prism that is most distant from the closest of the example hyperplanes.

larger, and the version space (i.e., the feasible space for separating hyperplanes) gets narrower, the solutions obtained using L_1, L_2, and L_∞ norms will be closer and closer to each other. See Fig. 3 for a visualisation of this behaviour.

Obviously, on the limit when the version space collapses to a single point, the three solutions are necessarily identical.

5.2. Version space

In Fig. 4 we can visualise what happens in a two-dimension version space. With L_2 norm, we maximise the distance between the example hyperplanes and a point in the surface of the cylinder where $\|\boldsymbol{w}\|_2 = 1$. For the L_1 norm, the points such that $\|\boldsymbol{w}\|_1 = 1$ define a prism with oblique faces, and we want to find the point in the surface of this prim that maximises the distance to the training examples. The L_∞ norm case is identical, except that the prism faces are parallel to the axes. In this figure we can see that the solutions for the support vector machines are likely to coincide if the version spaces are small. We can also see that when the optimal solutions (those farer from the closest "example" hyperplane) are close to edges of the prisms, important differences between the solutions are likely to happen.

6. Experimental results

We have tested the performance of the models using benchmark problems commonly referred to in the literature. The data for these tests, and their description, are available in (Blake and Merz, 1998).

The breast cancer database (below referred to as *breast*) was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg (Bennett and Mangasarian, 1992). Another test is the heart disease diagnosis test *heart*, which was collected in the Cleveland Clinic Foundation (Detrano et al., 1989). The *sonar* test corresponds to the data set used by Gorman and Sejnowski in their study of the classification of sonar signals using a neural network (Gorman and Sejnowski, 1988). The task is to train a network to discriminate between sonar signals bouncing off a metal cylinder and those which bounced off a roughly cylindrical rock. The last test *votes* concerns data of votes for each of the US House of

Representatives Congressmen, originally described in (Schlimmer, 1987).

6.1. Algorithm's performance

For testing the algorithm we first tried to maximise margins by linearly separating the data (i.e., admiting that the data are separable and using hard margins). If this problem is impossible (meaning that the data are non-separable), we run the algorithm again, this time using soft margins. (In practice, hard margins were used only for the sonar test, as all the other tests where not linearly separable.) We then checked the separation performance with the optimal hyperplane obtained this way.

We have used 10-fold cross-validation: each data set was randomly divided into 10 disjoint subsets with identical size; 9 of these subsets randomly chosen were then combined for training, and the remaining one for testing. This process was then repeated, and averaged over 10 trials. The performances obtained for each norm are reported in Table 1. As expected from the geometry of the distances, the results are not significantly different.

As said before, training the support vector machines for L_1 norm and L_∞ norm distances can be done using linear programming. These

Table 1
Average separation performance of support vector machines using different norms

		Breast	Heart	Sonar	Votes
L_2 norm	Train	97.154	82.684		94.684
$C = 0.01$	Test	**97.850**	78.796		94.059
L_2 norm	Train	97.138	84.665		96.013
$C = 0.1$	Test	97.567	81.419		95.650
L_2 norm	Train	97.218	84.665	**100**	97.419
$C = 1$	Test	97.567	81.419	92.857	96.337
L_2 norm	Train	97.250	84.665		**97.675**
$C = 10$	Test	97.567	81.419		**97.262**
L_2 norm	Train	97.250	84.702		97.547
$C = 100$	Test	97.567	81.419		**97.262**
L_1 norm	Train	96.932	83.895		95.553
$C = 0.01$	Test	97.710	79.785		94.730
L_1 norm	Train	97.059	**84.995**		97.163
$C = 0.1$	Test	97.282	81.419		97.019
L_1 norm	Train	**97.266**	84.555	**100**	**97.675**
$C = 1$	Test	97.567	**81.742**	**93.333**	**97.262**
L_1 norm	Train	97.250	84.738		97.598
$C = 10$	Test	97.567	81.419		**97.262**
L_1 norm	Train	97.250	84.702		97.624
$C = 100$	Test	97.567	81.419		**97.262**
L_∞ norm	Train	96.391	76.852		95.784
$C = 0.01$	Test	96.277	73.129		95.650
L_∞ norm	Train	97.075	84.005		95.784
$C = 0.1$	Test	97.567	80.441		95.650
L_∞ norm	Train	97.218	84.665	**100**	97.444
$C = 1$	Test	97.567	**81.742**	92.381	**97.262**
L_∞ norm	Train	97.202	84.445		97.572
$C = 10$	Test	97.567	81.419		**97.262**
L_∞ norm	Train	97.250	84.702		97.598
$C = 100$	Test	97.567	81.419		**97.262**

Results in bold face correspond to the machines with the best empirical performance. (The sonar problem is linearly separable, and hence the machine performance does not depend on the parameter C.)

Table 2
Comparison of CPU times for training the support vector machines using the MINOS optimiser, as a percentage of the L_2 norm machine CPU time

	Breast (%)	Heart (%)	Sonar (%)	Votes (%)
L_2 norm	100.0	100.0	100.0	100.0
L_1 norm	76.2	51.2	9.4	73.2
L_∞ norm	80.2	51.2	17.2	65.4

optimisation problems are numerically much easier to solve than the quadratic optimisation program corresponding to the L_2 norm distance machine. Even though the benchmark problems used are not too large, we could observe a significant improvement on the training times when we used L_1 and L_∞ norms, as shown in Table 2. The improvement is especially striking in the *sonar* case; because the data lead to a rather unstable instance, the quadratic optimisation problem is quite difficult to solve, but the linear optimisation problems are still easily solved.

We have used the MINOS software (Murtagh and Saunders, 1978) as the optimiser for all the cases. This is a solver for constrained, non-linear optimisation problems. Notice, though, that for the L_1 and L_∞ case, a specialised solver for linear programming is expected to lead to very significant improvements. We have solved problems with several tens of thousands of training examples in less that one minute in an Alpha Station machine, using a primal-dual interior point method for solving the linear program.

7. Conclusion

We have considered the L_1 and L_∞ norms, instead of the currently used L_2 norm, for training support vector machines. Based on the geometrical properties of each of these norms, we could propose that the results obtained by them should be rather similar; the hyperplane obtained with L_2 norm should somehow be "in between" the solutions for L_1 and L_∞ norms. In terms of separation performance, L_1, L_2 and L_∞ norm-based support vector machines tend to be quite similar, and no striking difference between them has been observed in our tests.

We introduced two new formulations for training support vector machines, allowing training for L_1 and L_∞ norms to be done with linear programming. This way, the separation problem becomes a much easier problem to tackle, as shown by the improvements on the training time in the results presented.

An extension of the methods proposed in this paper for kernel-based support vector machines can be done if we preprocess the data, and we input them to the training step after a non-linear transformation equivalent to the kernel transformation. This, however, may increase too much the dimension of the optimisation problem, and hence studying a formulation of kernel-based support vector machines where the optimisation problem is a linear program remains as a direction for future research.

Appendix A. Hölder inequality

Consider two real numbers $p, q > 0$ satisfying $1/p + 1/q = 1$ (Eq. (6)). For arbitrary $a, b > 0$ the inequality

$$\log\left(\frac{1}{p}a^p + \frac{1}{q}b^q\right) \geqslant \frac{1}{p}\log a^p + \frac{1}{q}\log b^q = \log(ab)$$

holds because of the convexity of the logarithm function. Hence

$$ab \leqslant \frac{1}{p}a^p + \frac{1}{q}b^q \tag{A.1}$$

holds for $a, b \geqslant 0$. Note that when $a = 0$ or $b = 0$ the inequality is trivial, and that equality holds if and only if $a^p = b^q$.

Let $\boldsymbol{u} = (u_1, \ldots, u_d)$ and $\boldsymbol{v} = (v_1, \ldots, v_d)$ be two d-dimensional vectors in $\mathbb{R}^d$. We define their inner product as

$$\boldsymbol{u} \cdot \boldsymbol{v} = \sum_{i=1}^{d} u_i v_i. \tag{A.2}$$

Using the p-norm definition of Eq. (4), $\|\boldsymbol{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p\right)^{1/p}$, we can write

$$\begin{aligned}\frac{|\boldsymbol{u}\cdot\boldsymbol{v}|}{\|\boldsymbol{u}\|_p\|\boldsymbol{v}\|_q} &= \frac{|\sum_{i=1}^{d} u_i v_i|}{\|\boldsymbol{u}\|_p\|\boldsymbol{v}\|_q} \\ &\leqslant \frac{\sum_{i=1}^{d}|u_i||v_i|}{\|\boldsymbol{u}\|_p\|\boldsymbol{v}\|_q} \\ &\leqslant \sum_{i=1}^{d}\left\{\frac{1}{p}\left(\frac{|u_i|}{\|\boldsymbol{u}\|_p}\right)^p + \frac{1}{q}\left(\frac{|v_i|}{\|\boldsymbol{v}\|_q}\right)^q\right\} \\ &= \frac{1}{p}+\frac{1}{q} = 1. \end{aligned} \tag{A.3}$$

Note that the first inequality in (A.3) becomes an equality when $\operatorname{sgn} u_i = \operatorname{sgn} v_i, \forall i$, and that the second inequality is due to Eq. (A.1). Eq. (A.3) implies the Hölder inequality, which states that:

$$|\boldsymbol{u}\cdot\boldsymbol{v}| \leqslant \|\boldsymbol{u}\|_p\|\boldsymbol{v}\|_q. \tag{A.4}$$

In this equation, equality holds when

$$\operatorname{sgn} u_i\left(\frac{|u_i|}{\|\boldsymbol{u}\|_p}\right)^p = \operatorname{sgn} v_i\left(\frac{|v_i|}{\|\boldsymbol{v}\|_q}\right)^q, \quad i = 1,\ldots,n.$$

The inequality can be easily extended to the case of $p = 1$ and $q = \infty$, as

$$\begin{aligned}|\boldsymbol{u}\cdot\boldsymbol{v}| &= \left|\sum_{i=1}^{d} u_i v_i\right| \\ &\leqslant \sum_{i=1}^{d}|u_i||v_i| \leqslant \sum_{i=1}^{d}|u_i|\max_j|v_j| = \|\boldsymbol{u}\|_1\|\boldsymbol{v}\|_\infty.\end{aligned}$$

References

Bennett, K.P., Bredensteiner, E.J., 2000. Geometry in learning. In: Gorini, C., Hart, E., Meyer, W., Phillips, T. (Eds.), Geometry at Work, Mathematical Association of America.

Bennett, K.P., Mangasarian, O.L., 1992. Robust linear programming discrimination of two linearly inseparable sets. Optim. Meth. Software 1, 23–34.

Bradley, P.S., Mangasarian, O.L., 1998. Massive data discrimination via linear support vector machines. Mathematical Programming Technical Report 98-05, University of Wisconsin, Computer Sciences Department.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2.

Blake, C., E. K., Merz, C., 1998. UCI repository of machine learning databases. URL http://www.ics.uci.edu/~mlearn/MLRepository.html.

Detrano, R., Janosi, A., Steinbrunn, W., Pfisterer, M., Schmid, J., Sandhu, S., Guppy, K., Lee, S., Froelicher, V., 1989. International application of a new probability algorithm for the diagnosis of coronary artery disease. Amer. J. Cardiol. 64, 304–310.

Gorman, R.P., Sejnowski, T.J., 1988. Analysis of hidden units in a layered network trained to classify sonar targets. Neural Networks 1, 75–89.

Murtagh, B.A., Saunders, M.A., 1978. MINOS 5.5 user's guide. Technical Report SOL 83–20, SOL, Stanford University, Palo Alto, CA, revised: July 1998.

Schlimmer, J.C., 1987. Concept acquisition through representational adjustment. Ph.D. Thesis, Department of Information and Computer Science, University of California, Irvine.

Smola, A.J., Frieß, T.T., Schölkopf, B., 1998. Semiparametric support vector and linear programming machines. NeuroCOLT2 Technical Report Series NC2-TR-1998-024, GMD.

Vapnik, V.N., 1995. The Nature of Statistical Learning Theory. Springer, Berlin.